National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

University of Alberta

The Leaky Window: A Congestion Control Technique for High-Speed Wide Area Networks

by

Charles Kiprotich arap Chirchir

A thesis
submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree
of Master of Science

Department of Computing Science

Edmonton, Alberta
Fall 1992

National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-77317-0

Canada

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Charles Kiprotich arap Chirchir
TITLE OF THESIS: The Leaky Window: A Congestion Control Technique for High-Speed Wide Area Networks

DEGREE: Master of Science
YEAR THIS DEGREE GRANTED: 1992

Permission is hereby granted to UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis not extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(Signed) .....

Permanent Address:
P. O. Box 1037,
Kericho,
Kenya.

Date: 8/6/92

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled **The Leaky Window: A Congestion Control Technique for High-Speed Wide Area Networks** submitted by **Charles Kiprotich arap Chirchir** in partial fulfillment of the requirements for the degree of Master of Science.

. . . . . .
Supervisor: Dr. A. Kamal

. . . . . .
External: Dr. W. Grover (Dept. of Elect. Eng.)

. . . . . .
Examiner: Dr. J. Harms

. . . . . .
Chair: Dr. T. A. Marsland

Date: June 4/92

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled **The Leaky Window: A Congestion Control Technique for High-Speed Wide Area Networks** submitted by **Charles Kiprotich arap Chirchir** in partial fulfillment of the requirements for the degree of Master of Science.

. . . . . .
Supervisor: Dr. A. Kamal

. . . . . .
External: Dr. W. Grover (Dept. of Elect. Eng.)

. . . . . .
Examiner: Dr. J. Harms

. . . . . .
Chair: Dr. T. A. Marsland

Date: June 4/92

Abstract

Congestion control in high-speed wide area networks (HS-WANs) is a very important issue since the presence of congestion may have catastrophic results. Reactive mechanisms applied to existing slow speed networks are not entirely suitable for HS-WANs due to the relatively long propagation delay. On the other hand, preventive mechanisms which have been proposed as possible solutions to this problem assume that traffic charateristics are known at the time of call setup. Window mechanisms do not make such an assumption, but modifications that have been proposed to adapt it to HS-WANs fall short of the fast response required in HS-WANs.

The Leaky Window (LW) mechanism is proposed in this thesis as an attempt to solve this problem. This mechanism is a modification of the sliding window that permits users to transmit traffic in excess of their window sizes based on an estimate of the network load. The estimate is based on acknowledgments received within a fixed time interval. Excess traffic is distinguished by "marking" cells. Marked cells are discarded at congested nodes. Conges-

tion control therefore, is a local decision executed by the congested node.

Through the use of a simulation model, it is shown that the LW has an average end-to-end delay and probability of loss that is lower than the sliding window mechanism. Comparison to the Virtual Leaky Bucket (VLB) mechanism shows that at lower load (0.7), the VLB has an end-to-end delay and a probability of loss that is lower than that of the LW under the same conditions. At higher loads where congestion is a real problem, the performance of the LW is significantly better than that of the VLB.

Acknowledgements

I thank my supervisor, Dr. A. Kamal for his seminal contribution, guidance throughout my research and for carefully reading my thesis drafts. I also thank the members of my examining committee, Dr. W. Grover and Dr. J. Harms for their positive criticism and comments which have been incorporated in this thesis. Special thanks to Dr. M. MacGregor for his comments.

Much thanks go to members of my family who although being thousands of kilometers away, have been a constant source of inspiration. I also express my gratitude to the Kenya/CIDA GTF project which supported me financially throughout my M.Sc. studies.

And finally, I thank all the people that I have interacted with, socially and academically, that made all the difference to the otherwise long and tiring hours of research.

Contents

1 Introduction 1

1.1 Introduction to networks 1

1.2 Wide Area Networks and B-ISDN 3

1.3 Protocol data units 7

1.4 Congestion in B-ISDN 8

1.5 Thesis objectives 11

1.6 Thesis outline 12

2 Literature Review 14

2.1 Preventive mechanisms 16

2.1.1 Bandwidth enforcement 16

2.1.2 Congestion control techniques by priority discarding . . 28

2.1.3 Resource management for traffic integration 35

2.2 Reactive mechanisms 40

2.3 Admission control 56

2.4 Summary 61

3 Leaky Window **64**

3.1 Introduction 64

3.1.1 ATM cell structure 65

3.1.2 Principles of operation 67

3.1.3 Router operation 73

3.1.4 User operation 76

3.1.5 Receive cell 78

3.2 Router Operation 80

3.2.1 Feedback signal generation 81

3.2.2 Feedback filtering 82

3.2.3 Congestion control 84

3.3 User operation 84

3.3.1 Window size adjustment frequency 85

3.3.2 Signal filtering 91

3.3.3 Window size adjustment and estimation of excess capacity 91

3.3.4 Duration of the idle period 94

3.4 Summary 96

4 Simulation model and results **98**

4.1 Description of the model 98

4.1.1 Definition of terms used 99
4.1.2 Message generation 101
4.1.3 Frame transmission 103
4.1.4 Receive cell 106
4.1.5 Receive acknowledgment 110
4.1.6 Router activity 112
4.2 Network Architecture 116
4.2.1 Dynamic window mechanism 122
4.2.2 VLB 122
4.2.3 Network specification 124
4.3 Results 126
4.3.1 Response to change 151
4.3.2 Fairness. 154
4.4 Summary 158

5 Conclusions and future research **161**

List of Figures

2.1 Functional diagram of a LB 18

2.2 Algorithm executed by a node 20

2.3 Model of a multi-hop virtual circuit with propagation delays and window control 45

2.4 Averaging cycle 53

3.1 ATM cell structure 68

3.2 PTI field modified when congestion is encountered 68

3.3 PTI field remains unchanged when no congestion is encountered 68

3.4 States of router processes 74

3.5 States of user processes 76

3.6 User receiver state diagram 79

3.7 Adaptive averaging 83

3.8 Window adjustment space time diagram. 87

4.1 Message generation 102

4.2 Strategy A frame transmission algorithm 107
4.3 Strategy B frame transmission algorithm 108
4.4 Receive cell and make ack when frame is complete 111
4.5 Receive acknowledgment 113
4.6 Estimate excess and adjust window size routines 114
4.7 Router receiver 117
4.8 Window update routines 118
4.9 Router transmitter 119
4.10 Congested Node 119
4.11 Three node network 120
4.12 Probabilty of loss for LW and Window mechanisms , for B_T= 200, Load =0.8 128
4.13 Comparison of probability of loss for LW and VLB for B_T= 100, Load = 0.7 133
4.14 Comparison of probability of loss for LW and VLB for B_T= 200, Load = 0.7 133
4.15 Distribution of duration of congestion for LW-A and VLB-A for B_T= 100, Load =0.7 134
4.16 Distribution of end-to-end delay for LW-A and VLB-A for B_T= 100, Load =0.7 135
4.17 Distribution of duration of congestion for LW-B and VLB-B for B_T= 100, Load = 0.7 136

4.18 Distribution of end-to-end delay for LW-B and VLB-B for B_T= 100, Load = 0.7 137
4.19 Distribution of duration of congestion for LW-A and VLB-A for B_T= 200, Load =0.7 137
4.20 Distribution of end-to-end delay for LW-A and VLB-A for B_T= 200, Load =0.7 138
4.21 Distribution of duration of congestion for LW-B and VLB-B for B_T= 200, Load = 0.7 139
4.22 Distribution of end-to-end delay for LW-B and VLB-B for B_T= 200, Load = 0.7 139
4.23 Comparison of probablity of loss for LW and VLB for B_T= 200, Load = 0.8 145
4.24 Comparison of probablity of loss for LW and VLB for B_T= 300, Load = 0.8 145
4.25 Distribution of duration of congestion for LW-A and VLB-A for B_T= 200, Load = 0.8 146
4.26 Distribution of duration of congestion for LW-A and VLB-A for B_T= 300, Load = 0.8 147
4.27 Distribution of duration of congestion for LW-B and VLB-B for B_T= 200, Load = 0.8 147
4.28 Distribution of duration of congestion for LW-B and VLB-B for B_T= 300, Load = 0.8 148

4.29 Distribution of end-to-end delay for LW-A and VLB-A for B_T= 200, Load = 0.8 148
4.30 Distribution of end-to-end delay for LW-A and VLB-A for B_T= 300, Load = 0.8 149
4.31 Distribution of end-to-end delay for LW-B and VLB-B for B_T= 200, Load = 0.8 149
4.32 Distribution of end-to-end delay for LW-B and VLB-B for B_T= 300, Load = 0.8 150
4.33 Througput variation with time for LW and VLB for B_T= 200, Load =0.8 152
4.34 Mean queue length variation at router over time for LW and VLB; B_T = 200, ρ = 0.8 153
4.35 End-to-end delay variation with time for LW and VLB for B_T= 200, Load =0.8 153

List of Tables

4.1 Rate based on cells and frames per credit-interval (VLB) . . . 123

4.2 Summary of results for the Window and LW mechanisms for $\rho = 0.8$ . . . 128

4.3 Summary of result for LW for $\rho = 0.7$ . . . 131

4.4 Summary of result for VLB for $\rho = 0.7$ . . . 132

4.5 Summary of results for LW for $\rho = 0.8$ . . . 141

4.6 Summary of results for VLB for $\rho = 0.8$ . . . 143

4.7 Througput, delay for nodes with propagation delay τ and 2τ, $B_T = 200$ and $\rho = 0.7$ . . . 155

4.8 Througput, delay for nodes with propagation delay τ and 2τ, $B_T = 300$ and $\rho = 0.8$ . . . 156

4.9 Througput, delay for nodes with propagation delay τ and 2τ, distingiushed credits at $B_T = 200$ and $\rho = 0.7$ . . . 156

4.10 Througput, delay for nodes with propagation delay τ and 2τ, using distinguished credits, $B_T = 300$ and $\rho = 0.8$ . . . 157

Chapter 1

Introduction

1.1 Introduction to networks

Despite its many facets, the term "Networking" collectively means "to link together for the purpose of sharing." Used in the context of computers, networking is the configuration of transmission facilities to serve a number of geographically dispersed users. The transmission facilities are shared for efficient utilization. Such sharing is achieved by using one of a number of switching techniques.

Circuit switching is one of those methods in which a complete path of connected links is set up before the start of communications. In this method of switching, the links in the established circuits remain connected for the entire duration of the communication session. At the end of the session, the

connection is torn down. Time division multiplexing (TDM) and frequency division multiplexing (FDM) are other methods of switching. In FDM, the frequency spectrum is divided among the communicating sessions, with each session having exclusive possession of its frequency band. TDM may be synchronous or asynchronous. In synchronous TDM one or more time slots is dedicated to each communicating session. Asynchronous TDM does not allocate channels permanently to communicating sessions. Instead, channels are allocated to active sources on demand. This technique, therefore, introduces overhead required to identify the channel user.

In message switching, complete messages that include a header, the users data and a trailer are forwarded by the switch on a link after accumulating the whole message. Message switching is usually done in a store and forward manner. Packet switching is similar to message switching except that in the former, long messages are segmented into smaller blocks. As in message switching, a header and a trailer are attached to each block.

Economical connectivity is the primary goal behind the employment of switching. To achieve such an economy, a compromise between the communication link cost and the complexity of the switching system must be struck.

Different network classifications exist. For example, networks may be classified according to geographical coverage, the method of switching employed, the services provided, topology or the operational strategy.

Geographical classification categorizes networks into Local Area Networks (LANs) that usually cover an area of a few kilometers, Metropolitan Area Networks (MANs) that cover an area equivalent to a metropolis or Wide Area networks (WANs) that may span thousands of kilometers.

The choice of switching method classifies networks as either circuit switched, message switched. or packet switched. Hybrid networks which employ combinations of packet and circuit switching may also exist.

In LANs and MANs, several nodes may share a the same communication medium. Medium Access Control (MAC) strategies therefore, must be used in LANs and MANs to resolve and avoid contention. The IEEE 802 committee has standardized several MAC protocols for LANs [59] and MANs [41]. Some of the commonly used LAN protocols are the Ethernet, the Token ring and FDDI [53]. Reference [1] surveys the existing LANs.

1.2 Wide Area Networks and B-ISDN

Wide area networks in form of point-to-point connections over telephone lines have existed since the early 1950's. This was motivated by the advent of digital computers, and the resulting military and commercial interest in large scale data processing. The availability of cheap and powerful microprocessors in the past two decades brought about migration from large centralized systems towards smaller autonomous machines. With the increased use of

autonomous machines, the advantages of having an interconnection facility became apparent.

As LANs provided the capabilities to interconnect computers over a limited geographical area, the need to interconnect those LANs and their component machinery increased. The Internet was built as a response to interconnect the many LANs which were geographically dispersed. It is a packet switch WAN that is based on the X.25 protocol. The emergence of high-speed LANs and workstation, and the growing role of supercomputers in scientific computing have led to a new, largely unfulfilled requirement for high speed computer communications. Such devices and applications that require bandwidth higher than that provided by the traditional slow speed networks have been the impetus for the development of high speed WANs (HS-WAN). HS-WANs are expected to provide a variety of services. This include interconnection of intelligent nodes requiring large bandwidth, interconnection of isolated networks, and applications such as distributed processing, full motion video (e.g., HDTV) and computer imaging (e.g., seismic, medical and weather), all of which require large bandwidth, in addition to applications that are adequately served by existing slow speed networks.

The decreasing cost of optical fiber and advancement in laser technology have been the vehicle for the rapid development of high speed networks. To incorporate the diverse requirements mentioned above, broadband integrated services digital networks (B-ISDN) concept has been conceived as an all

purpose digital network. The network capabilities which B-ISDN is expected to support include [39]:

-Interactive and distributive services

-Broadband and narrowband rates

-Support for bursty and continuous traffic

-Connection-oriented and connectionless services

-Point-to-point and multi-point communications

There are still a number of problems that must be addressed before high speed networks can be implemented.

WANs typically consist of arbitrary topologies with a large number of interconnections. The interconnections are made via switches. The nature of traffic that such switches must handle cover a wide spectrum, as indicated by the variety of services to be provided by HS-WANs. Each traffic type may have different requirements. To meet these requirements, several switching and multiplexing techniques (transfer modes) have been proposed for B-ISDN. These schemes include the circuit-switching based synchronous transfer mode (STM) and the packet-switching based asynchronous transfer mode (ATM). The STM technique was initially considered an appropriate transfer mode for B-ISDN because of its compatibility with existing systems. In STM the transmission capacity is organized into periodic time frames which are themselves comprised of time slots. Each call is assigned a particular slot within a frame. Thus, a call is identified by the position of the slot.

Assignment of time slots to calls is based on the peak transfer rate of the call so that the required quality of service (QOS) can be guaranteed even at peak load. An STM-based interface tends to be rigidly structured. To provide the capacity required by different services, multiple STM channels must be used. This may increase the complexity of the switching function [39].

In ATM, the usable capacity (bandwidth) is segmented into fixed-size information-bearing units called cells. The ATM cell consists of a 5 octet header, and a 48 octet information field. Cells are transmitted over a virtual circuit, and routing is performed based on the virtual circuit identifier (VCI) contained in the cell header. ATM's fundamental difference from STM is that slot assignments are not fixed; instead, the time slots are assigned in an asynchronous (demand-based) manner. By allocating bandwidth on demand, ATM has the advantage that it can easily be adapted to suit the requirements of different applications. ATM has therefore been chosen as the transfer mode for B-ISDN.

A layered architecture similar to the ISO OSI model has been defined for ATM. A discussion of the structure and functions of the ATM architecture is presented in [3].

1.3 Protocol data units

To transmit a message a network, it is usually represented as a string of binary symbols, that is one and zeros. This symbols are commonly referred to as bits. When transmitting these message bits across a network, control overhead in the form of additional bits must be added to ensure factors such as reliable communication, correct routing, and prevention of network congestion. In addition, transmitting long messages as one complete unit is generally not employed since this is detrimental to message delay, buffer management and congestion control. Therefore long messages are normally broken up into shorter bit strings.

Segmentation and reassembly of messages may be done according to the *Open System Interconnection* (OSI) reference model which has been standardized by the ISO. The OSI model consists of seven layers. The layers starting from the bottom are the physical layer, data link layer, network layer, transport layer, session layer, presentation layer and the application layer. The following are protocol-data-units at each layer. The physical layer is concerned with transmitting raw bits stream over a communication channel. The data link layer breaks input data up into data frames. At the network layer, data is represented in packets with header containing the routing address added to the packet received for the transport layer. The transport layer segments messages into packets and reassembles packets into

messages. The remaining higher layers process complete messages.

In the OSI model, messages are segmented into packets at the transport layer are sent through the network as individual units and are reassembled into complete messages at the destination station. The sending data link layer module places overhead control bits at the beginning and the end of each packet, resulting in a longer bit string referred as a *frame*.

In the ATM architecture, the adaptation layer accepts packets from higher layers and adds a header and a trailer to form frames. Frames are further segmented into 48 octet blocks before being submitted to the ATM layer. At the ATM layer, a header is added to form a 53 octet ATM cell.

1.4 Congestion in B-ISDN

The phenomenon of congestion has been well known in packet switched data communication networks since their inception [6]. In such networks, bursty sources are statistically multiplexed to gain bandwidth efficiency. Even though the sum of the average rates of these sources is kept less than the bandwidth of the link they are multiplexed on, the instantaneous aggregate rates of packet generation can exceed this bandwidth. Buffers are provided to queue packets during such intervals; however, congestion which is defined as the visible degradation of QOS seen by the communicating sessions [20] may result if the durations of these periods of "overload" are long enough, such

that packets are lost when the limited amount of buffer space is exhausted. Congestion can also occur because of the failure of network components; for example, traffic rerouted due to failure of a common link can cause a sudden rise in the traffic intensity in other links.

The congestion control problem in broadband networks is considerably different from that in traditional packet-switched data communication networks. Some important differences are:

- The presence of a wide range of traffic in broadband networks, each with its own traffic characteristics, QOS and performance requirements. Traffic to be supported include those applications requiring bandwidth of a few kilobits (e.g., slow terminal) per second to those that require several hundreds megabits per second (e.g., moving image data). Traffic characteristics also vary with applications such as interactive data and video which is highly bursty while some traffic, like large file transfers, tend to have a continuous nature. B-ISDN is also expected to meet the diverse service and performance requirements of multi-media traffic.

- Broadband speeds make it harder to engineer efficient feedback mechanisms to quench sources to relieve congestion. For example, suppose two nodes, A and B, are 4,000 kilometers apart. An electromagnetic signal launched at node A would take 20 milliseconds to propagate to node B. Consider a data network that uses 56 kilobits/second links be-

tween the two nodes. The transmission time of a 53-byte packet is 7.57 milliseconds. Thus when the first bit of the first packet transmitted by node A arrives at node B, node A would be transmitting the third packet. If a node B is congested and signals node A to stop sending, about six packets would be on their way to node B, before node A shuts itself off. No packet would be lost if six extra packet buffers were available at node B. If the network used 45 megabits/second links, the transmission time of a packet would be 9.42 microseconds. Suppose node A is sending data to node B. It takes 9.42 microseconds to fully inject each 53-byte packet into the transmission link. If node A is transmitting continuously, the first bit of a packet will arrive at node B while node A has transmitted another 2,123 packets.

Thus if node B signals node A to stop send, about 4,246 packets would be on their way before node A could respond. Therefore, using a 45 megabits/second, congestion is more serious than in a data communication network based on 56 kilobits/second links where feedback control is used. At higher speed this problem is even more pronounced.

- It is recommended that the layers of flow control should be reduced into as few flow control mechanisms as possible [38]. This will minimize the processing at intermediate nodes, which is a major consideration in high speed networks. Layer 2 data link control procedures are not

terminated at intermediate nodes in broadband networks but instead, only a relay function is performed at these nodes. Thus, link-by-link controls which would take advantage of the smaller propagation delay between adjacent nodes of a network cannot be applied.

- Software based mechanisms are slow and dedicated hardware could be expensive. Therefore, algorithms to control congestion should be as simple as possible.

Congestion control in B-ISDN has therefore received increasing attention in the past few years.

1.5 Thesis objectives

A major consideration of congestion control mechanisms in high speed networks is simplicity. Part of the success of the sliding window mechanisms (will also be referred to as window mechanisms) in narrowband networks it's simplicity. Such mechanisms achieve flow control objectives and efficient bandwidth utilization by employing statistical multiplexing of traffic sources. Window control mechanisms which are reactive, control the rate of transmission of traffic sources based on feedback from the network. The presence of a relatively longer propagation delay in HS-WANs renders the existing reactive control mechanisms inadequate. We would like a congestion control

mechanism that retains the desirable features of window mechanisms, and at the same time, overcomes the drawbacks of reactive controls in broadband networks.

A congestion control mechanism based on the window with these objectives is proposed in the thesis. The proposed mechanism modifies the window mechanism by permitting sources to transmit traffic in excess of their window size based on an estimate of the load in the path of the connection. Sources are assumed to transmit traffic in the form of ATM cells. Cells that are in excess of the window size are "marked," and may be discarded at any congested node in the path of a connection.

1.6 Thesis outline

The rest of this thesis is organized as follows. In Chapter 2, we survey congestion control mechanisms that have been proposed for HS-WANs. Such schemes are classified as preventive or reactive controls. Admission control which is considered a preventive mechanism is presented on its own, as it is applied at a higher level to both preventive and reactive schemes. At the end of Chapter 2, an analysis of the existing congestion controls is presented, and the congestion control mechanism proposed in this thesis, namely, the Leaky Window (LW), is briefly introduced. In Chapter 3, the LW window mechanism is proposed and its operation is discussed in detail. In Chapter

4, the simulation model used to test the LW and performance results using this model are presented. Finally in Chapter 5, we present the conclusions and future research directions.

Chapter 2

Literature Review

In this chapter techniques that have been proposed for flow and congestion control will be reviewed. *Flow control* is an agreement between a source and a destination to limit the flow of packets without taking into account the load in the network. *Congestion control* is primarily concerned with controlling the traffic to reduce the overload on the network. Flow control limits traffic based on buffer availability at the destination, whereas, congestion control limits traffic based on buffer availability at the intermediate nodes [35]. Congestion control can be either preventive or reactive. Reactive controls react to congestion after it happens and try to bring the degree of the network congestion to an acceptable levels. Preventive controls on the other hand, try to prevent congestion before it happens. The objective of preventive controls is to ensure *a priori* that the network traffic will not reach the level which

causes unacceptable congestion [3].

Congestion control mechanisms which do not strictly ensure that congestion is prevented but instead distinguish cells or frames that may be discarded if an unacceptable level of congestion is reached, are included in the class of preventive mechanisms. Discarding techniques applied at nodes (routers, switches or multiplexors) are also categorized as preventive mechanisms.

Resource management schemes are used to enable the integration of diverse traffic types with different performance requirements. These schemes focus on the management of bandwidth, buffers and, uses appropriate scheduling and queueing algorithms at nodes to minimize congestion. We classify resource management schemes as preventive.

Admission control mechanisms can be preventive or reactive, depending on the strategy used. However, admission control operates on a time scale of call duration, which is longer than the time scale of operation of the preventive and reactive congestion control mechanisms mentioned above [6]. Admission control may therefore be applied in addition to preventive and reactive congestion control mechanisms that operate on a shorter time scale. We shall therefore mention issues addressed by admission control and some of the admission control strategies without specifying them as preventive or reactive.

In this chapter, preventive and reactive control mechanisms for congestion control will be reviewed first. The controls mechanism reviewed include those

proposed for implementation at the transport layer and therefore described in term of packets while others are to be implemented at the ATM layer and therefore described in terms of cells. The actual layer at which the mechanism is to be implemented is not mentioned in the discussion, but the level of implementation will be clear from context. We shall also discuss some of the criteria and admission control techniques that have been proposed.

2.1 Preventive mechanisms

Preventive mechanisms may be subdivided further into short and medium term flow and congestion control mechanisms according to the time scale of operation of the mechanism. Short term mechanisms will be categorized into bandwidth enforcement and congestion control by priority discarding schemes. Medium term flow and congestion control mechanisms are generally integrated resource management schemes, and will also be presented in this section.

2.1.1 Bandwidth enforcement

(i) Stop and go congestion control

This method of flow control avoids congestion by employing a per connection packet admission policy at the edge of the network, and a particular service

discipline at switching nodes, namely the stop-and-go queueing. Central to both parts of the strategy is the notion of time frames, hence the name *framing strategy*. Starting from a reference point in time common to all nodes, the time axis is divided into periods of some constant length T, which are referred to as frames. A time frame of size T seconds during which no more than rT bits are admitted is referred to as (r, T) smooth. The admission policy requires that for a connection k once set up and a transmission rate r assigned, it has packet arrivals to the network which are required to be (r, T) smooth.

This strategy ensures bounded delay at each node and hence a bounded delay on any path. Bounded delay is achieved at the cost of the strict admission policy enforcing the smoothness property on packet arrivals, which may result in added delay. Low utilization of the transmission capacity may result if the framing strategy is uniformly applied to all services in a broadband network.

Appropriate combination of the framing strategy with other traffic management schemes such as the inclusion of a category of users whose traffic is not sensitive to delay, may be employed. The traffic from these low priority users can be transmitted when there is no high priority traffic scheduled for transmission. This allows low priority traffic to benefit from statistical multiplexing, while the higher priority users continue to enjoy guaranteed services of the framing strategy. The analysis in [24] shows that such features can

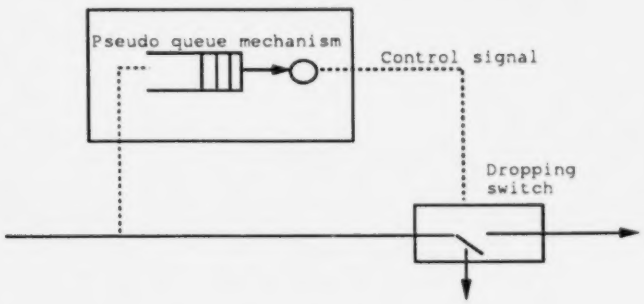

Figure 2.1: Functional diagram of a LB

be implemented without violating the negotiated rates. The impact of frame size T on queueing delay and bandwidth allocation is also discussed in [25].

(ii) The Leaky Bucket (LB)

The LB may be implemented at the cell or packet level. The following discussion is the cell level implementation of the LB. A functional diagram of the LB mechanism is shown in Figures 2.1. The source peak bandwidth, B_p, is first checked and enforced by dropping cells when their inter-arrival times are smaller than a minimum value calculated from the declared value of B_p. The mean bandwidth, B_m, the burstiness, b, which is equal to B_p/B_m, and the average burst length, L, are enforced as follows. A counter is incremented with every cell arrival, and is decremented at a rate B_e. If a cell arrives when the counter has reached the maximum value Q, it is considered to be in excess of the peak rate and is dropped. If either $B_e = B_p$ or $Q = \infty$, no enforcement on B_m or L applied. On the contrary, if $B_e = B_m$ and $Q = L$,

then the utilization factor of the pseudo-queue is too high, which may result in an unacceptable cell dropping rate [22].

Simulation studies [22] show that a fraction of cells, which is determined by the source characteristics, is dropped before entering the network. This is because the LB is a policing mechanism and the number of cells dropped depends on the policing parameters used. If $B_e = B_m$ and $Q = L$ as mentioned above, a high rate of discarding would result.

Further analysis of the LB has been presented in [7, 50, 56]. In [50], the LB is analyzed as a $G/D/1$–s queue. When the service time in the model is equal to the mean cell inter-arrival time, the offered load is 100%. In this case, a very high counter limit would be required to obtain an acceptable cell loss probability. Although the realization of a high counter limit is not very difficult, the dynamic behavior of the mechanism becomes very poor, because it takes too long to detect a parameter violation. This problem may be solved by decrementing the LB counter faster than the mean cell inter-arrival time. The counter limit required for a given cell loss probability decreases because the offered load of the corresponding queueing model is below 100%. Numerical results show that a counter limit of 50 is sufficient to obtain a probability of cell loss of 10^{-10}, if the counter is decremented in intervals corresponding to 80% of the mean cell inter-arrival time[50]. Thus, the dynamic behavior of the system improves significantly, but the source may exceed the negotiated mean cell rate by 25%.

```
repeat
  wait_event(incoming cell)
  if (buffer is not full)
    queue_arriving_cell(buffer)
  else if (buffer full)
    if (cell is marked)
      drop_cell
    else if (marked cell in buffer)
      remove_marked_cell(buffer)
      queue_arriving_cell(buffer)
    else
      drop_cell
until(forever)
```

Figure 2.2: Algorithm executed by a node

The virtual leaky bucket (VLB) [22] is a modification of the LB that allows cells that exceed the negotiated rate into the network. The VLB uses the same pseudo-queue mechanism as the LB shown in Figure 2.1 and those cells recognized as excess cells are transmitted into the network after being marked instead of being dropped as done in the LB mechanism. Therefore all cells enter the network and are dropped by the network nodes if and only if congestion develops. To implement the VLB mechanism, the "dropping" function in the scheme of Figure 2.1 is substituted by a "marking" function, and the network nodes manage the cells in the buffer according to the algorithm shown in Figure 2.2. For the same offered load, it is shown by simulation that the percentage of cells that are dropped is reduced by employing the VLB [22]. This is because unlike the LB, the VLB mechanism overcomes

merging of policing and congestion control by marking excess cells instead of dropping them. Discarding of marked cells at congested nodes rather than at the network interface as is done in the LB, is a local decision based on the state of the congested node at the arrival time of each cell.

Another variant of the LB is the generalized leaky bucket GLB [4]. The GLB is to be implemented at the packet level, unlike the LB or the VLB which may be at the cell or packet level. We shall therefore discuss the operation of the GLB in terms of packets. This scheme proposes to use packets marked red and green, where the green packets are transmitted at the rate guaranteed at the time of call setup, while the red packets represent the rate in excess of the negotiated rate. Red and green packets are processed at the intermediate nodes according to traffic conditions in the network using one of the following policies:

Policy # 1. The total number of red packets at a node cannot exceed a threshold B_T.

Policy # 2. The total number of packets at a node cannot exceed a limit B_T. Both green and red packet are allowed to enter the node until the total number of packets at the node equals the threshold B_T. When the number of packets exceeds B_T, any red packets that arrives at a node is discarded.

Simulation studies show that the second strategy has a performance (through-

put against threshold) closer to the ideal case (green packets dropped only when all buffers are filled by green packets) for a Poisson arrival process of traffic at an intermediate node. The bursty nature of data makes exact estimation of traffic requirements and parameters difficult.

The LB has received a fair amount of analysis [4, 7, 22, 50]. The low multiplexing efficiency makes this method an unlikely candidate for practical use in congestion control of HS-WANs. The VLB and the GLB flow control techniques are improvements over the LB and may be more suitable in high speed WAN. Below, we shall discuss the reasons why the VLB may be more suitable for HS-WANs. In the VLB,

1. Sources do the policing without dropping cells. Instead, cells are marked as eligible for discarding at congested nodes. Higher statistical multiplexing can therefore be achieved, and

2. Each node can act autonomously, discarding cells only when congestion occurs. This is an attractive aspect, since it minimizes the amount of information exchanged by nodes during congestion which may otherwise lead to *congestion collapse* [1] [23].

The challenging problem is to determine the policing parameters [4, 50]. Using either of these schemes requires foreknowledge of the traffic charac-

[1] Decrease in useful throughput caused by an increase of offered load beyond the critical system capacity.

teristics of each source. Policing parameters may be assigned according to traffic classes such as those suggested in [28]. This however, is done at a higher level and not the cell level where the VLB operates. The issue of reallocation of resources when some calls terminate may be addressed as to whether the policing parameters should be static or dynamic. A problem that the VLB or the GLB does not address is the priority of transmission. This would be required at the intermediate nodes in order to determine the order in which cells arriving within the same slot are transmitted or which cells to discard if unmarked cells arrive at a congested node.

(iii) Peak counters

The peak counter [42] input rate control is a mechanism where the decision to mark or delete a packet is based on the period of time that an input source has been operating above its nominal average rate. The mechanism is implemented with two counters. The first counter is used to compute the average behavior of the input source. The second counter keeps track of how long the first counter is kept above its threshold, i.e. how long the input source has been operating above the nominal rate. The range of the second counter is between zero and a maximum value T_{pk}. Its value is increased at a constant rate while the average counter is above the threshold; otherwise, it is decreased at the same constant rate. Cells that arrive while the source has been operating beyond the threshold value longer than some maximum value,

(T_{pk}), are marked. There are two variations of the peak counter approach. In the first one, which will be referred to as PC1, the average counter is not limited but marked cells are not counted. In the second, which we will refer to as PC2, the average counter has a maximum value Q, and marked cells are still counted. The performance of both variations of the peak counters scheme is compared to the LB. In [42], simulation results show that PC2 introduces the smallest probability of loss on well-behaved sources. Reaction time is defined as the time it takes a mechanism to go from a predefined state to the marking state. A comparison of the reaction time of the LB and the peak counter mechanisms is performed under the following assumptions. In order to obtain a lower bound, cells are assumed to arrive at the peak rate until the marking state is reached. Reaction time for the LB is found to be linearly dependent on the bucket size. For a fixed buffer size of 50K (cells), the reaction time of the LB is the smallest followed by PC1 and then PC2.

(iv) Virtual Clock

The Virtual Clock flow control mechanism is proposed in [67]. Data transmission by a user i is referred to as a $flow_i$. At each switching node or gateway, a data flow is assigned a Virtual clock ($Vclock_i$), whose step ($Vtick_i$), is equal to the mean inter-packet gap, $1/AR_i$, where AR_i is the average transmission rate indicated in the request. Each switch performs the following set of operations:

1. During the setup of a $flow_i$ request, compute $Vtick_i$.

2. Upon arrival of the first packet from $flow_i$, set $Vclock_i$ to the value of real time.

3. For each arriving packet from $flow_i$, advance $Vclock_i$ by $Vtick_i$, and stamp the packet with the value of $Vclock_i$.

Packets are transmitted in increasing order of $Vclock$ stamp values at a node. If, on arrival of a packet, the buffer is full, the packet with largest value of $Vclock$ is dropped from queue. To police each flow, an inspection period AI_i is set. Each $Vclock_i$ is inspected at AI_i intervals for violation of the negotiated rate. Whenever $flow_i$ exceeds a threshold T, a warning message is sent to the source.

If a burst of packets arrives from $flow_i$ that has been idle for a while (with an AI_i period), the burst can cause sudden queueing increases to other flows. This is because the algorithm is designed to tolerate flow variations within each average interval (which is primarily chosen by individual application). For a flow that has been idle, $Vclock_i$ will not be advanced after the last checking point, until $AI_iR_i(= AR_i \times AI_i)$ packets have been received. This conflict is resolved by assigning an auxiliary clock $auxVC_i$ to each flow. The algorithm presented above is modified to use $auxVC_i$ as follows. Upon receiving each packet from $flow_i$,

1. $auxVC_i \leftarrow \max(\text{real time}, auxVC_i)$

2. $Vclock_i \leftarrow (Vclock_i + Vtick_i)$, $auxVC_i \leftarrow (auxVC_i + Vtick_i)$

3. stamp the packet with the value of $auxVC_i$.

Flow monitoring Upon receiving every set of AI_iR_i data packets from $flow_i$, the switch checks the flow in the following way:

1. If ($Vclock_i$– real time) > T , a warning message should be sent to the flow source.

2. If ($Vclock_i$ < real time), $Vclock_i \leftarrow$ real time.

It is difficult to choose proper values of threshold T, such that whenever ($Vclock_i$– real time) > T, the switch can assume that $flow_i$ has indeed been transmitting too fast. The difficulty arises because of variation of the flow's data generation over each average interval which intuitively should cancel each other out and hence, the $Vclock$ reading would stay within some vicinity of real time. However, the variation is found to grow unbounded [67]. A user-behavior envelope is therefore used to regulate the behavior of sources so that a meaningful value of T may be set. The user-behavior envelop is a window that permits flows to transmit AI_iR_i packets in an average interval.

Transmission of packets by the order of time stamps at intermediate nodes ensures that each flow is serviced according to its priority. For example, flows of higher rates or priority get preferential treatment. A the same time, lower priority packets are not denied a chance for service completely. Simulation

results [67] show a high link throughput for homogeneous traffic. For diverse throughput flows, different flow rates have a minor impact on the average queueing delay. Lower rate flows experience a higher queueing delay. The reason is that the virtual clock ticks for the lower throughput flows are bigger than for a higher throughput flow. A packet arrival in a lower throughput flow causes the $Vclock i_i$ to be advanced by a quantity that is much greater than that of a higher throughput flow. A packet arrival in a lower throughput flow will therefore have to wait for one or more packets from higher-throughput flows to be transmitted before the next scheduled transmission time of the lower throughput flow.

The above phenomenon has also been observed in [22]. It indicates that the GOS seen by different flows depends on the network load. Results in which users transmit faster than the negotiated rate show that this technique detects violating sources in the event of congestion [67].

This scheme incorporates flow and congestion control as well as priority of service. However, it entails time stamping and priority handling, all of which are computationally intensive.

2.1.2 Congestion control techniques by priority discarding

(i) Optimal Discarding

In [46], the effects of the choice of packets to be discarded in the event of congestion are investigated. A queueing system associated with an outgoing link of a switch or statistical multiplexor in which the primary queueing occurs at the output link is considered. The service interval is a constant value τ, corresponding to packets of a constant length and a constant bit rate communication link. During a service interval, up to L packets may arrive at the queue. The paper analyses the case in which packet arrivals are not correlated and the arrival process is stationary. The arrival statistics can therefore be reduced to a set of probabilities as follows. D levels of packet priorities are defined. p_d is the probability that an arriving packet is of priority $d, 1 \leq d \leq D$, and q_l is the probability of l arrivals in a service slot τ, $0 \leq l \leq L$. The header of each packet contains a delivery priority. A cost function $c_d, 1 \leq d \leq D$ is associated with discarding a packet of class d. The discarding policy is a control decision per service interval and is based on the queue size and delivery priorities. All packets in the waiting queue may be discarded, but not those already in service, *i.e.* those at the heads of their queues.

Consider the case of a queue of size N with B as the number of buffers

occupied by packets awaiting service. l is the number of packets that arrive while the server is busy. Before the server goes into service, it discards $N-B$ packets if $N > B$, according to the discarding policy, so that B packets are left in the queue. The number of possible states, S, is therefore $\sum_{i=0}^{N-1} D^i$. Each state s_i has an associated discard policy which results in a specific discard action. The number of policies can be large even for moderate values of (N, L, D). A single policy which minimizes the expected cost per arrival is used, which is referred to as *optimal discarding.*

In [46], optimal discarding is compared to the default policy of discarding only arriving packets. Let δ_d be the probability of discarding a packet of priority d. The expected cost per arrival can be expressed as

$$E(c) = \sum_{d=1}^{D} \delta_d p_d c_d$$

It is demonstrated that the optimal policy has a lower $E(c)$ than the default policy when the network is overloaded. For example, comparison of the maximum load that can be sustained by the default policy and the optimal policy without exceeding $E(c) = 0.35$ shows that the optimal policy can sustain a larger maximum load than the default policy. In this comparison, a larger maximum load indicates a better performance.

To apply this discarding technique to HS-WANs, the number of states S must be kept small to minimize the computation required to determine which packets should be discarded. This condition is further emphasized by

the fact that the discard policy computation must be done at the beginning of each service slot.

(ii) Bit dropping

In [57] the smoothing effects of bit dropping during congestion on the superposition of traffic in a packet voice multiplexor is investigated. Bit dropping is justified by the fact that voice is tolerant to bit dropping to a certain extent. Although speech exhibits a high degree of burstiness, when the less significant bits in voice packets are dropped during states of congestion in the multiplexor, significant smoothing of the packet voice process is achieved by speeding up the packet service rate during critical periods of congestion in the queue [57]. By this argument, the superposition of packet arrivals can be viewed as a Poisson process. This model is analyzed as an $M/\dot{D}/1/K$ queue, where $\dot{D}$ denotes the deterministic but state dependent nature of packet service times.

A 2-phase burst/silence model voice generator is used in simulation studies of this bit dropping scheme [57]. Two arrival patterns with mean speech talkspurt and silence lengths of 532 and 650 milliseconds respectively (35% activity) and 420 and 550 milliseconds (42% activity) respectively, are tested. Simulation results of the mean queue length and the mean delay match closely those obtained analytically. The packet loss fraction is generally negligible up to fairly high number of active voice sources. Mean delay (in milliseconds)

for the case without bit dropping is about an order of magnitude higher than the corresponding mean delay for the case of bit dropping at a link utilization of about 95%.

(iii) Selective packet discarding

Reference [65] proposes to selectively discard packets from arrival streams at a packet voice multiplexor during periods of congestion. Digitized voice packets are classified into two classes. A fraction, α, of the packets are in Class 1 and the rest, $1-\alpha$, are in Class 2. Classes 1 and 2 can be considered to contain the most and the least significant information respectively. The classification of packets can be determined by:

1. Embedded coding: this technique groups encoded information into more and less significant bits.

2. Even/odd sample: speech samples are identified as either odd or even, the odd samples being put in Class 1 and even samples in Class 2 or vice versa.

3. Speech energy detection threshold: the energy in a segment of speech corresponding to one packet is estimated and compared to each of two thresholds. If it lies below both thresholds, no packet is transmitted. If it exceeds the lower threshold but not the higher one, the segment

is classified as a "semi silence" and is placed in Class 2. Speech with energy exceeding both thresholds is placed in Class 1.

During congestion Class 2 packets are discarded. Two methods used to detect the onset of congestion are:

1. Speaker activity: When the number of active callers exceeds a threshold T, incoming Class 2 packets are discarded.

2. Buffer content threshold: When maximum queue length exceeds Q_{max}, Class 2 packets are discarded on arrival.

Performance analysis of three congestion control schemes is done with congestion thresholds chosen so as to obtain approximately the same blocking probabilities for the following control strategies:

vc1 : Packet classification by embedded coding or even/odd samples, and a fraction, β, of Class 2 packet arrivals from each source is discarded.

vc2 : Packet classification by speech energy threshold method and all Class 2 packets are discarded during congestion.

vc3 : Packet classification done using embedded coding or by even/odd samples, and newly arriving Class 2 packets are discarded when buffer occupancy reaches the threshold Q_{max}.

Numerical results obtained for control strategies $vc1, vc2$ and $vc3$ show that when the voice traffic is not heavy (less than 48 voice sources), $vc2$ performs better than the other two strategies in terms of the mean waiting time (milliseconds) and blocking probabilities. However, as the voice traffic further increases, $vc3$ has a better performance in terms of mean waiting time than $vc1$ and $vc2$.

Another scheme proposed in [66] uses a bivariate model as in [65] to analyze two discard methods:

1. Instant discard of arriving packets when buffer is full.

2. Random selection of packets to be discarded when congestion occurs.

In this scheme, discarding of packet is done only when there is an overflow.

Let $S_n = (t_n, b_n)$ be the system state at the beginning of the n^{th} slot, where t_n is the number of voice users in a talkspurt and b_n is the queue length. Let a *slot* be equal to the packet transmission time and γ be the number of slots in a checking interval. M is the buffer size. For discarding strategies 1 and 2, overflow may occur under the following conditions:

1. $b_n \geq \gamma$ and $t_n > M - b_n + 1$.

2. $b_n < \gamma$ and $t_n > M - b_n + 1$.

The probability of discarding is different for the two schemes which results in different queue lengths after the discarding process.

Numerical results obtained using a two phase speech generator show that the larger the delay constraint, the smaller the discard probability for the two strategies. The mean discard probability as a function of the number of active voice sources of the first scheme is slightly lower than that of scheme 2, but the difference is very small. For instance, a talk spurt in scheme 2 never looses more than 11 packets but there is a 0.2% chance to loss more than 11 packets in the first scheme. For an average packet loss probability of 0.01, the distribution of the number of lost packets per talkspurt indicates that scheme 2 has fewer talkspurts that incur loss than the first scheme. Scheme 2 has a lower variance of the number of discarded packets per talkspurt than scheme 1. for example, the lowest value of variance for scheme 2 is 0.008 while it is 0.03 for the first scheme. For increasing number of active voice sources, the variance of scheme 2 is always less than that of scheme 1.

(iv) Congestion control for real-time traffic in high speed networks

In [54], a method for congestion control for real time traffic is proposed and analyzed. The scheme discards packets according to one of the following criteria:

- Discard packets if the time spent at the node under consideration exceeds the end-to-end deadline, d.

- Discard a packet if the time to be spent in the current node i exceeds a fixed local deadline τ_i.

These congestion control techniques are analyzed using a FIFO bounded system time (BS) model and a FIFO bounded waiting time (BW) model. A study of total loss composed of packet dropped at intermediate nodes and packets missing their end-to-end deadline as a function of the local deadline is done in [54]. Analytical and simulation results for a network with 5 nodes and a throughput of 0.8 show that the FIFO-BW dropping scheme reduces total losses from 5% to 3.1%. Drop losses incurred by tight local deadlines are not compensated for by reduction in downstream traffic. It is also shown that for very tight deadlines, almost all packets that have not been dropped at intermediate nodes make their deadlines. A comparison between heterogeneous and homogeneous traffic deadlines show no significant changes in packet loss.

2.1.3 Resource management for traffic integration

(i) Fratta's proposal

Fratta [51] has proposed an input regulation scheme that guarantees traffic parameters through a shaper device and then optimizing the bandwidth assignment for required grade of service (GOS). The GOS is obtained through a suitable bandwidth assignment based on the parameters guaranteed by the

shaper, rather than that enforced by the policer. The physical parameters of a shaper are set at call setup based on traffic characteristics of the user. The shaper is composed of a server operating at a bit rate B_s and a window mechanism that allows the transmission of at most m cells in a time interval of D slots. The window permits transmission of traffic with mean bit rate equal to min $[B_p m/D, B_m]$, where B_p and B_m are the peak and the mean bandwidth, respectively, and the maximum burst length is limited to the value m. The output of the shaping device has a peak bit rate equal to min $[B_p, B_s]$. To minimize cell loss, the shaper parameters have to be chosen so that $B_p m/D \geq B_m$.

(ii) Bandwidth management scheme by Dighe et al.

The scheme proposed in [12] effectively combines datagram and virtual circuit traffic. Two priorities, *lo* for low priority packets and *hi* for high priority packets are used. The 2-queue system is based on the observation that if the amount of circuit-like traffic (*hi-queue*) can be predicted, an allowance can be made for it when statistical packets (*lo-queue*) are scheduled. At a node (gateway or switch) packets are served according to priority by scheduling packets in the *lo* and *hi* queues. A time of scheduling is associated with the scheduling of each *lo* priority packet. Thus when a *lo* priority packet is scheduled for service, an allowance for the *hi* priority packets is made. The allowance is based on the number of packets awaiting service at the

output queue and the expected number of arrivals of *hi* priority packets. The number of packets awaiting service indicates the amount of delay that an arriving high priority packet will incur before being transmitted.

An access node with four types of input traffic is used to test this strategy. The input traffic patterns used are:

1. A source generating fixed length messages with an exponentially distributed inter-arrival time transmitted as datagrams (D).

2. A virtual circuit data source with exponential message arrival distribution and a bursty message length distribution (VC1).

3. A virtual circuit data source with bursty arrivals of fixed length messages (VC2).

4. Circuit traffic whose packet arrival process is uniformly distributed (CT).

When the 2-queue method is applied, CT traffic is placed in *hi-queue* while the rest of the traffic types defined above are placed in the *lo-queue*. The following service strategies are applied at a node:

1. Rate control with priority to CT traffic and all the other traffic being treated as virtual circuit traffic.

2. Rate control with the 2-queue method and all types of traffic being treated as virtual circuit traffic.

3. Rate control with 2-queue scheduling of packets. Here, VC1, VC2 and CT are given virtual circuit treatment and D is given datagram treatment. That is, packets from D are made to join the shortest queue.

Simulation results show that the 2-queue strategy is effective in giving high priority to the CT traffic with a minimum of delay jitter. The 2-queue strategy with rate control and alternate routing of datagram packets is better than all the other strategies in the following performance measures: mean and standard deviation of packet delay distribution in the output queue and message delay distribution. At the expense of a slightly longer average delay for CT traffic, the throughput of datagram traffic is doubled. A similar or better performance is obtained for virtual circuit data sources and the maximum delay for data traffic is bounded. This is achieved without giving up the jitter requirement of CT traffic.

(iii) A flow control scheme by Hac

In the flow control scheme proposed in [28], traffic is classified into three classes. Class 1 consists of isochronous circuit-like traffic that has a guaranteed low bounded delay by the network. Class 2 is non-isochronous traffic with modest peak data rates where individual virtual circuits are capable of using only a small fraction of the total capacity of the transmission facilities in the network. Finally, Class 3 is for traffic which has high peak data rates and is bursty. In this class, a single VC may transfer data to the full capacity

of internal links of the network. Classes 1 and 2 are allocated guaranteed bandwidth.

The network has the ability to completely shut off any bursty data sources or at least delay their packet transmissions at the nearest node if congestion occurs. When a buffer threshold is exceeded, then either the source or the nearest node directly sending packets to this point is shut off. Buffer capacity adequate to accommodate packets sent during the shut off period is provided at potential points of congestion. It is shown analytically that the elapsed time of packet transfer at a switch increases as the number of packets buffered and the shut off time increase. The results show that this congestion control algorithm should be used in networks in which the propagation delay from the congested node to shut off node is short. This however, contradicts the view that node-by-node flow control may not be a very suitable flow control technique for HS-WAN [38]

(iv) Duration-limited statistical multiplexing (DLSM)

The scheme proposed in [26] is composed of 3 elements. Call admission, and packet admission are imposed at the edge of the network, and a stop-and-go queueing service discipline at the switch nodes. During call admission, a call is set up based on availability of resources to support the desired grade of service and a set of traffic regulations for the call. A call may be associated with a set, Ω, of priority classes of the network. For each connection k, loss

free transmission of traffic with priority $\omega \in \Omega$ is achieved by allowing an aggregated length of admitted packets with priority $\omega \in \Omega$ to be limited to $r_k^{\Omega} T$ bits, where r_k^{Ω} is the regulation parameter assigned to connection k subject the constraint that

$$\sum_{k \in l} r_k^{\Omega} \leq C_l, \qquad l = 1, ...l$$

where l is the number of links in the path and C_l is the capacity of the link l. The stop-and-go queueing discipline is used at switch nodes. The end-to-end delay is bounded, and loss free transmission is guaranteed only to the highest priority class. Other priority classes experience a degree of packet loss that increases as the priority goes down.

2.2 Reactive mechanisms

Reactive mechanisms rely on feedback from the network in order to adjust the traffic input in response to the current state of the network. Because of the relatively long propagation delay in HS-WANs compared to slow speed networks, reactive mechanisms are slow in responding to network congestion in HS-WANs. Most of the mechanisms in this category are based on the window mechanism. Congestion control of data transmission using the window mechanism is well established. This is discussed in [23, 33, 48, 67] and in many books dealing with communication networks such as [5, 55] and [63].

In a conventional window mechanism, the window size W is negotiated at the connection setup. A source node is permitted to have a maximum of W unacknowledged packets outstanding. The two main objectives of congestion control are to strike a reasonable compromise between low delay and large throughput and still maintain fairness for all users. The end-to-end fixed window strategy is not satisfactory in either respect [5]. What is really needed in end-to-end window flow control to achieve a good delay-throughput tradeoff is dynamic adjustment of window sizes. Such proposals have been made in [19, 30, 40, 48].

In this section, reactive mechanisms that have been proposed for high speed networks will be reviewed. The methods that are used for window adjustment will be presented first.

(i) Analysis of the increase and decrease algorithm for congestion avoidance in computer networks.

Reference [10] analyses different increase/decrease methods for the window flow control mechanism. The methods considered are a combination of multiplicative and additive increases and decreases of the window size. These schemes are compared on the following criteria:

Efficiency measures resources utilization,

Fairness measures the allocation of resources to the different users. The *maxmin* fairness criterion has been widely adopted [21, 34, 47]. Essentially, the set of users is partitioned into equivalence classes according to which resource is their primary bottleneck. The *maxmin* criterion asserts that users in the same equivalence class ought to have equal shares of the bottleneck. If we denote the load of the i^{th} and the j^{th} users by $x_i(t)$ and $x_j(t)$, respectively, then a system is operating fairly if $x_i(t) = x_j(t), \forall i, j$ for users sharing the same bottleneck.

Distributedness indicates status information is required by the method to make control decisions.

Convergence measures the time it takes for the system to reach equilibrium.

To satisfy distributedness, convergence, efficiency and fairness, it is shown analytically that the decrease policy should be multiplicative and the linear increase policy should always have an additive component. Optionally, it may have a multiplicative component with a coefficient that is not less than 1. Nonlinear adjustment is not pursued for reasons of parameter sensitivity and computational complexity. DECBIT, the flow and congestion control technique proposed in [36] uses an explicit feedback, and the window size adjustment mechanism is based on additive increase and multiplicative decrease. This scheme is shown to have small amplitude oscillations. The

control method is also shown to be responsive to changes in network load. However, these tests are performed in slow speed networks where propagation delay is not as significant as in HS-WANs.

(ii) Fairness and congestion control on a large ATM data network with dynamic adjustable windows.

Reference [30] proposes a window control scheme with the objectives of fairness, efficiency and stability. At a node, the buffers is divided into a number of buffers, one for each virtual circuit, together with some unassigned memory space. Traffic arrivals at a node from each virtual circuit are queued in the buffer assigned to it. The per circuit queues are served in a round robin fashion. Routers enforce edge-by-edge cell windows on virtual circuits. Each virtual circuit is assigned a default window at connection set up, and it retains a window of at least this size for as long as the circuit exists. The input router may request an increase or a decrease in the window size. A maximum of a full round trip window may be assigned. Increases in window sizes may be requested at intervals as short as one round trip. A request travels around the virtual circuit and the routers to either confirm or modify it at the same time making changes to the respective buffer allocation. The router may begin to use a larger window after the "increase" buffer allocation request has been confirmed. Window size allocation for new connections is based on available bandwidth and the activity of the existing connections. The rene-

gotiation timer is a timer that monitors the duration of idle intervals. When a communicating session has been idle for some predefined duration of time, the renegotiation timer triggers a reduction in window size. The duration of the idle time could be as short as one round trip time; however, the network will probably allow the communicating sessions a few seconds of idle time before reducing their windows.

This scheme may achieve stability and fairness depending on how requests for additional bandwidth are allocated. Efficiency may however be harder to realize with this scheme because of the dependence of the buffer allocation procedures on the round trip propagation delay and the limits imposed on the user idle times.

(iii) Dynamic adaptive windows for networks with multiple paths

References [19, 40] propose an algorithm for dynamic window control for high speed networks which have multiple links, as shown in Figure 2.3. The links may have different propagation delays. Only one connection consisting of M nodes is fully depicted in Figure 2.3, and cross-traffic from other connections which is also referred to as exogenous cross traffic, are shown as incomplete paths. The theoretical treatment assume that the network in Figure 2.3 is balanced, *i.e.*

$$\lambda = r_1 - \nu_1 = r_2 - \nu_2 = ... = r_m - \nu_m$$

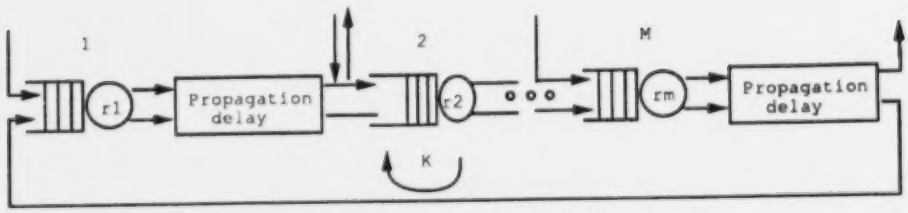

Figure 2.3: Model of a multi-hop virtual circuit with propagation delays and window control

where at node i, r_i is the service rate and ν_i is the exogenous cross traffic rate. The parameter λ is called the common *reduced nodal processing time* for the virtual circuit. M is the number of hops and K is the window size. For a window of size K, there are three basic operating regimes

1. $K/\lambda \ll 1$: light usage
2. $K/\lambda \sim 1$: moderate usage
3. $K/\lambda \gg 1$: heavy usage

The design or choice of values of K and λ is done with a desire to optimize source *power*[2]

It is shown analytically that the *power* increases linearly at light usage with increasing K, when all other parameters are constant. At heavy usage,

[2]The power of any resource is defined as: $power = \frac{throughput^{\alpha}}{responsetime}$, $where : 0 < \alpha < 1$. $\alpha = 1$ used here, maximizes power at the knee of response time curve, which is the point at which the increase in throughput is small while the response time increases rapidly with load.

power is inversely proportional to K, when all other parameters are held constant. Hence, power is maximized in the moderate usage regime.

The main results obtained by maximizing the power with respect to the window size K are given below. These are asymptotic solutions for large λ. The superscript "*," indicates optimizing values.

1. $K^* \sim \lambda - \alpha^* \sqrt{\lambda}$

2. $N_1^* \sim \beta \sqrt{K^*}$

3. $\sigma^*(N_1) \sim \gamma \sqrt{K^*}$

where N_1 and $\sigma(N_1)$ are the steady state mean and standard deviation of the number of packets of the virtual circuit which are queued at node 1. These statistics are common to all queueing nodes, and $\alpha^*, \beta^*, \gamma^*$ are universal constants since they depend neither on λ nor on K (and are therefore oblivious to the details of the physical world).

Adaptive windowing: A framework for adaptive windowing based on the above theoretical results gives the following equation which will be referred to as the design equation [40]:

$$(R^* - 1)\sqrt{K^*} \sim M\beta^*.$$

This design equation is a property of the asymptotic optimal design and expresses the desired relation between R (the mean round trip response time), which can be established from observation, and K (the window length), which

is to be controlled. The quantity $(R^* - 1)\sqrt{K}$ is a monotonically increasing function of K for all fixed λ.

The problem of adaptively estimating K^* from the design equation in a stationary environment with a fixed buffer size, where the value of λ is unknown and the number of hops, M, is fixed and known, is of the form $\psi(K) = C$, where C is a given positive number and $\psi(K)$ is a regression function.

An algorithm for solving this type of problem by stochastic approximation is given in [52], which, in this case, is of the form:

$$K_{n+1} = K_n - a[(R_n - 1)\sqrt{K_n} - b],$$

where n indexes the packets acknowledged at the source. The gain parameter a affects responsiveness to changing exogenous conditions as well as long-term variability of the window around its optimal value.

In data networks, the residual processing rate λ is not only unknown, but is also non-stationary. Since K^* changes with λ, it is necessary for the algorithm to track K^*. This is achieved by a choice of design constants which compromise the opposite requirements implied by tracking capability and a good behavior in the stationary mode. Several variations of the algorithm have been proposed:

- Adaptive windowing based on measurements of the nodal buffer contents [40].

- Adjustment done less frequently than with the arrival of each acknowledgment to allow for averaging [40], and
- Hybrid algorithms as suggested in [48], where K is incremented additively and decremented multiplicatively.

The adaptation algorithm is extended to the case where multiple circuits share a common path. It has been shown in [67] that some adaptive windowing algorithms lead to:

- Large amplitude oscillations in the queues,
- Unfair treatment of virtual circuits with long paths.

Simulation results for a 45 Mbit/sec link with a round trip propagation delay of 47 milliseconds show no large amplitude oscillatory behavior in either the windows or the nodal packet queues in this scheme.

This scheme responds to network traffic much faster than schemes that accumulate acknowledgments, but suffers from the fact that feedback from the destination is still required in order to varies the window size.

(iv) Tri-S Scheme

Tri-S, the scheme proposed in [64], addresses the problem of oscillations in the window size due to the additive increment/multiplicative decrement method

of window size adjustment [33]. Traffic load is deduced from acknowledgments by using the normalized throughput gradient (NTG). The throughput gradient, $TG(W_n)$, is equal to $\frac{T(W_n)-T(W_{n-1})}{W_n-W_{n-1}}$, where W_n and W_{n-1} represent two sequential window sizes and $T(W_n)$ is the throughput at window size W_n. $NTG(W_n)$ is equal to $\frac{TG(W_n)}{TG(W_1)}$. The basic-adjustment-unit (BAU) is the minimum size by which the window can be adjusted. A reasonably small value of BAU is recommended if the packet size is large so the adjustment of the window size can be fine and gradual.

Distribution of available transmission capacity, is done whenever there is a significant change of traffic in all connections. As an example, resource distribution is done when a new call is established and the overall traffic demand (or window size) can no longer be accommodated with the resources available and therefore results in the overflow of buffers. In such a case, all users start a new session of demand adjustment.

Three operation modes are defined. At connection set up, the window size is set to one (BAU). The window size is incremented on receipt of each acknowledgment until the maximum allowed size by the end user is reached. When a packet is timed out it is retransmitted and the user enters decrease mode. The window size is set to one BAU. Upon receiving an acknowledgment, the NTG is checked. The window size is increased by one BAU if NTG exceeds a threshold, NTG_d, otherwise it enters the increase mode. In the increase mode, the window is increased by *(BAU/current window size)* each

time an acknowledgment is received. If the accumulated increase is greater than the packet size, NTG is examined. If NTG is less than a threshold NTG_i, the window size is decreased by a packet size, otherwise no action is taken. The Tri-S scheme is found to have smaller amplitude oscillations compared to other schemes.

(v) Congestion avoidance mechanism in high speed transport protocols

In the high speed transport protocol proposed in [44], control packets that contain complete state information are exchanged between the transmitter and the receiver at intervals T_{IN}, equal to $\max[\frac{RTD}{k}, IPT]$, where RTD is the round trip delay. k is a constant and IPT (inter-packet time) is the average time between two data packet transmissions. If a logical connection is inactive since the last state transmission, T_{IN} is increased by a factor of 2 up to a maximum of $\max[\frac{RTD}{m}, IPT\ ]$. The information in a receiver state packet includes the number of buffers available at the receiver and the bit map representing the outstanding blocks that may have been transmitted but not acknowledged so far. The transmitter state packet contains the interval between the two state transmissions T_{IN}, the number of blocks queued for transmission and the number of all transmitted blocks (including those that have not been acknowledged). The queue length may be used for flow control within the network and admission control of new connections.

(vi) Distributed source control (DSC)

Distributed source control, the protocol proposed in [49], uses rate-based control managed by individual sources. At the connection setup, the window size W_s and smoothing interval T_s are negotiated. For a virtual circuit i, the DSC parameters W_{si} and T_{si} are chosen such that $\frac{W_{si}}{T_{si}} = \min(\lambda_i, \lambda_a)$, where λ_i is the requested average throughput during active periods and λ_a is the speed of the access link. The minimum value of $T_{si} \geq T_{min}$ (T_{min} is suitably chosen for the link speed) that yields an integer value W_{si} for the DSC window and satisfies the above equation is chosen. That is, if

$$\lambda_i = \frac{W_1}{T_1} = \frac{W_2}{T_2} = ..$$

with

$$T_1, T_2, ... \geq T_{min},$$

then the smoothing interval $T_{si} = \min(T_1, T_2, ...)$ and the DSC window $W_{si} = \lambda_i T_{si}$.

The window size W_s for any virtual circuit (VC) therefore, is the minimum of all the window sizes that are allocated by each of the links along the path. The smoothing period is similarly negotiated. The window size and the smoothing interval may be renegotiated again when resources become available. Simulation results [49], show that the number of buffers needed to achieve a given blocking probability is reduced when DSC is used in comparison to when it is not. Delay in non-prioritized systems decreases with

the use of DSC. For prioritized service, DSC has a negligible effect on delay. The variance of delay depends on values of W_s and T_s. The variance in the inter-departure interval is large ($\sigma^2 = 1.12$) when $W_s = 320$ packets and $T_s = 50$ milliseconds, and reduces to $\sigma^2 = 0.002$ when $W_s = 8$ packets and $T_s = 1.25$ milliseconds.

(vii) Explicit congestion notification in broadband packet networks

In the flow control modeled in [2] explicit congestion control notification is made to traffic sources in the event of congestion. To implement the control procedure, each packet contains a field in its header to convey congestion information, known as the explicit congestion notification (ECN). The ECN field is initially reset to zero, to indicate that the packet has not experienced any congestion. Intermediate routers, upon experiencing congestion, set the ECN field to one to indicate congestion. Defining W_{max} is the maximum window size and W_{eff} is the effective window size at time t of a connection. Packets are admitted into the network queue when the number of packets awaiting acknowledgment is less than W_{eff}, packets not instantaneously admitted to the network are queued at the access queue. W_{eff} is adjusted according to the network congestion status. At the network nodes, the average queue length is used as an indication of congestion at the node. The duration over which the average queue length is observed is of variable length and is related to the queue regenerative cycle as shown in Figure 2.4. At the

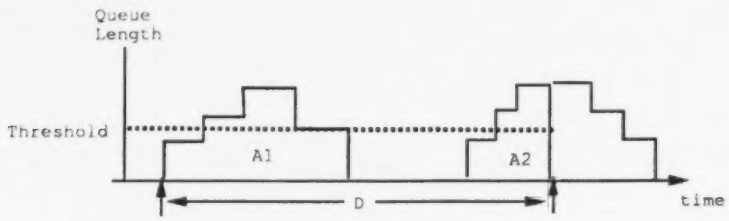

Figure 2.4: Averaging cycle

arrival instant of each packet the average queue length, Q is evaluated as

$$Q = \frac{A_1 + A_2}{D}$$

If Q exceeds the threshold B_T, the ECN in the acknowledgment packet is set. At the destination, the ECN field is copied into the acknowledgment packet and transmitted to the source. The acknowledgment packet suffers a propagation delay of τ to reach the source. The source accumulated the arriving acknowledgments until it equals the window size. The decision to increase or decrease the window size is based on a majority the sign of the ECN in the acknowledgments received. If 50% or more of the ECN received indicate no congestion, the window size is additively increased by one, otherwise, the window size is multiplicatively decreased by a window reduction factor.

Simulations results [2] of this scheme over DS1 and DS3 channels show that the probability of overflow increases with the buffer threshold. The window reduction factor used to reduce the window size during congestion has a marked effect on the probability of overflow. Using a window reduction

factor of 0.5 results in a probability of loss that is an order of magnitude smaller than that achieved with a window reduction factor of 0.875. In general, this scheme has a lower probability of overflow when compared to a fixed size window scheme. The fixed size window scheme however, has a lower end-to-end mean delay than the ECN scheme. This can be regarded as an example of the classical notion of trading delay for a higher throughput.

(viii) Versatile message transaction protocol (VMTP)

VMTP is a transport protocol described in [8]. Flow control is based on a packet group in which the transmitter is allowed to send a group of data packets in one operation– one VMTP message. An acknowledgment for the group is awaited. This scheme has advantages over the window mechanism in a client server system where requests for response such as files transfers are made. Once the client is ready to transmit, it transmits a chunk of data and waits for a response. Resources are therefore not tied down unnecessarily. However, the server must be prepared to accept the packet group at once. Transport protocols in the same category as VMTP include the express transfer protocol (XTP) and the network block transfer protocol (NETBLT) [14]. XTP merges the network and transport layers into a common transfer layer. Its design is built on many ideas from earlier protocols such as rate control, selective retransmission and implicit connection establishment. NETBLT was developed for high throughput bulk data transfer. It

is designed to operate efficiently over long-delay links such as those provided by satellites. Flow control is performed by means of windows and through the use of rate control, which limits the number of packet transmissions in a negotiated time interval. In XTP and NETBLT, rate control specifies the rate of transmission and the maximum burst size while in VMTP it specifies the inter-packet gap.

(ix) Loss-load curves

A loss-load curve characterizes the percentage of packets that may be dropped at a given client and network load, allowing the client to determine and control the delivered packet loss by controlling its packet rate [9]. Using the loss-load curve information, a host can choose its own tradeoff between throughput and packet loss. In this scheme, gateways monitor network load and provide explicit feedback to source hosts. Feedback is based on the loss-load curve, derived from observed traffic intensity and the capacity of the gateway. The probability of loss for each user is evaluated at a gateway based on the network load. The level of network load is estimated by packet counting and exponential averaging of the number of packets received over an interval of time. The gateway counts the number of packets received in a fixed interval of time. At the end of the interval, an average rate is computed for each sender based on the number of packets received in the interval. The gateway maintains an exponentially weighted average of sender rates over

the last several intervals. These average rates are used for loss-load curve updates, and for rate enforcement in the next interval.

Two parameters, a and b, measure the total (excess) traffic arriving at the gateway and the distribution of traffic among competing users, respectively. The parameters a and b are sent to traffic sources every averaging interval for loss-load curve updates. The averages are also used to enforce the rate of transmission in the next interval. The interval may be chosen to suit the sophistication, stability, responsiveness and the overhead of the scheme. In the presence of congestion, appropriate loss-load curves reduce the amount of loss-sensitive traffic in the network while making random inter-burst bandwidth available for less loss-sensitive traffic. A type of service field in the header allows the gateway to bias packet drop towards less loss-sensitive traffic for a source-destination pair. Since the loss-load curve is a function of the network load, the grade of service seen by a user is a function of the network load and the user's chosen imposed load.

2.3 Admission control

Admission control decides whether to accept or reject a new connection based on whether the required performance can be maintained. Reference [3] surveys some of the proposed admission control schemes. The major issues being studied in admission control are:

1. Traffic parameters (traffic descriptors) required to accurately predict network performance.

2. Criteria used by the network to decide when to accept a new connection.

3. Dependence of network performance on the various traffic parameters.

(i) Traffic descriptors

When a connection is requested, the network needs to know the traffic characteristics of the new connection in order to accurately predict its ability to maintain a certain performance level. The peak rate, the average bit rate and a measure of burstiness are among the most commonly used parameters for traffic descriptors. Among these traffic descriptors, burstiness is the most important [3]. However, no consensus has been reached on an appropriate way to describe the burstiness of a traffic source. Some definitions of burstiness include:

1. The ratio of peak bit rate to average bit rate [11, 13, 22],

2. The average burst length, i.e., the mean duration of the interval during which the traffic source transmits at the peak rate [31],

3. Cell jitter ratio defined as the variance-to-mean ratio of the cell inter-arrival times, or Var [cell inter-arrival times]/E[cell inter-arrival times] [32] ,

4. The squared coefficient of variation of the inter-arrival times, or Var [cell inter-arrival times]/E^2[cell inter-arrival times] [58] , and

5. Peakedness defined as the variance-to-mean ratio of the number of busy servers in a fictitious infinite server group [17].

(ii) Decision criteria

The transmission delay and the cell loss probability are good indicators of the degree of network congestion and are therefore the most commonly used decision criteria in admission control [3]. When transmission delays and cell loss probability are applied to admission control, their long-term time average values are usually used [11, 13, 31, 43]. Using long-term averaged values however, may not be sufficient in an ATM network because here, the network traffic changes rapidly and dynamically, forcing the network to move from one degree of congestion to another [37]. In [37], an instantaneous cell loss probability is proposed and used as a decision criterion to consider the temporal behavior of the network.

The decision to accept a new call should be based on the remaining unused bandwidth that may be allocated to new calls (residual effective bandwidth) and the effective bandwidth requirement of an arriving call [16]. The desirable features of an admission control scheme are:

- On line computation should be small.

- The amount of stored data required for on line computation should not be excessive.
- The approach should operate in terms of the effective bandwidth allocated for each call and the residual effective bandwidth available for new calls.
- If the approach is not close to exact in terms of GOS requirements, it should be conservative rather than optimistic. This feature ensures that the required GOS for calls already in progress and the calls to be accepted into the network can be met.

It is pointed out in [15] that a comprehensive congestion avoidance strategy and control plan consists of control policies that operate on various time scales that range from distributed real-time controls to long term network engineering procedures.

Preventive mechanisms such as the LB, VLB, GLB, stop and go, and the Virtual clock mechanisms are all rate based. These mechanisms require the traffic characteristics to be known at call setup time in order for the transmission rate or the policing parameters to be assigned to sources. Determining policing parameters is a challenging problem [4, 50]. These mechanisms can be made more dynamic by using resource management methods in order to take into consideration various traffic types and varying network conditions. Reallocation of bandwidth when new calls are accepted into the network or

when some calls terminate can be done by instructing active users to re-adjust their policing parameters using a strategy such as the loss-load curves [9].

Reactive mechanisms in general rely on feedback to control the rate of transmission. Knowledge of the traffic characteristics at call setup in this case is not necessary as the sources will adapt their rate of transmission according to network conditions. Due to relatively long propagation delay HS-WANs, reactive control mechanisms are not as effective as when applied to slow speed networks.

Window mechanisms have been applied successfully to traditional slow speed networks. However, in HS-WANs, the rate at which the window size may adapt to the network conditions is subject to the following delays:

1. Propagation and processing delays at intermediate and destination nodes.

2. Delay incurred while accumulating acknowledgments in order to take into consideration the effect of the previous window adjustment which is usually done in order to minimize oscillations of the window size.

This makes the response of a window mechanism unsuitable for HS-WANs because:

1. Using accumulated feedback to adjust the window size achieves the aims of flow control if the delay incurred (while accumulating feedback)

does not exceed the time that the traffic sources take to significantly change their transmission pattern. In a HS-WAN, traffic sources are expected to vary much faster than the end-to-end delay.

2. Depending on the maximum window size allowed, maximum statistical multiplexing of traffic in the network may not be realized.

3. Real time traffic that will be transmitted using HS-WANs has an upper bound on admissible end-to-end delay. Traffic from such an application may suffer heavy losses if the delay incurred while waiting for credits to become available is greater than the maximum delay that can be tolerated.

Attempts to make the window mechanism more suitable for HS-WANs have been made in [2, 19, 30, 40]. These however, still fall short of the fast response expected of a congestion control mechanism in the broadband regime because window adjustment is based entirely on feedback.

2.4 Summary

In this chapter, congestion control schemes that have been proposed for high speed networks were reviewed. The schemes are categorized into preventive and reactive controls. Preventive controls such as the LB, VLB, GLB, stop and go, and the Virtual clock are all rate-based mechanisms. Priority

discarding techniques that may be applied at nodes such as optimal discarding, bit dropping, random discarding and selective packet discarding were found suitable for applications like voice and video which can tolerate loss. Discarding mechanisms are also categorized as preventive controls.

Resource management for traffic integration utilizes basic mechanisms such as LB or stop and go, to control individual sources. The traffic control parameters assigned to the basic control mechanisms may be based on the traffic classification and the network load.

Reactive mechanisms in general rely on feedback to adjust the rate of transmission according to network conditions. The window mechanism is the most common in this category. The main differences between the window mechanisms reviewed in this chapter is in the use of received feedback and method of window adjustment.

We have also briefly reviewed the admission control techniques, which covers the policy of whether a new call should be accepted or not. Admission control employs traffic descriptors and decision criteria.

In the next chapter, we propose a modification to the window mechanism that improves the response time of the window mechanism by permitting a window size of unacknowledged frames to be outstanding and in addition allows traffic that is in excess of the window size to be transmitted. The excess frames are marked and are eligible for dropping at congested nodes. As in the other window schemes, the window size is adjusted periodically as a function

of accumulated feedback and/or the number of marked frames transmitted successfully. The number of marked frames transmitted from a source may be determined from the feedback and the window size which reflects the transmission capacity that may be utilized and the delay encountered over a connection.

Chapter 3

Leaky Window

3.1 Introduction

The flow and congestion control scheme proposed and studied in this chapter is referred to as the Leaky Window (LW). It is applicable to both connection oriented and connectionless services. In the ATM architecture, the adaptation accepts packets form higher layers and add a 5 octet header and a trailer to form frames. Frames are segmented into 48 octet block before being submitted to the ATM layer. At the ATM layer, a header is added to form a 53 octet ATM cell. We shall use these definitions of a frame and cell in the remainder of this thesis.

The LW mechanism is an end-to-end protocol which uses the window mechanism to control the number of unacknowledged frames that each source

may have outstanding in the network. Each traffic source performs the following operations:

1. Transmit data in ATM cell format,
2. Receive acknowledgments from the destination, and
3. Adjusts the window size.

The entity that performs the operations listed above is referred to as the *user*. The term *end-user* and user may be used interchangeably. Routing nodes will be referred to as *routers*. The destination will reconstruct frames from the received cells. Positive acknowledgments (PAK) are sent to the source when complete and error free frames are received by the destination, while negative acknowledgments (NAK) are sent for frames that are received in error. Transmission between routers is in the form of ATM cells. The LW mechanism may be implemented at the transport, the network or the adaptation layer. In this chapter, we shall discuss the principles of operation of the LW mechanism, and the operation of the routers and users in detail.

3.1.1 ATM cell structure

The ATM cell consists of a 5-octet header and a 48-octet information field [60, 61]. The CCITT header format which will be used at the User-Network-Interface (UNI) is shown in Figure 3.1. The header contains a 4-bit "generic

flow control" (GFC) field, a 24-bit label field containing the virtual path identifier (VPI) and virtual channel identifier (VCI), a 3-bit payload type (PTI) field, a 1-bit cell loss priority field (CLP) field, and an 8-bit header error check (HEC) field. At the Network-Network-Interface (NNI), the 4-bit GFC is used as part of the VPI field.

The GFC field is used to assist the customer premises equipment in controlling flow of traffic for different qualities of service. The exact procedures of how to use this field are not agreed upon as yet. The VPI provides an explicit path identification for a cell, while VCI provides an explicit channel identification for a cell. A virtual path is a bundle of virtual channels which can be switched as one unique channel. The PTI field can be used for maintenance purposes, and it indicates whether the cell contains user information or network maintenance information. When the cell contains user information, the PTI field is set to 000 or 001, for a cell that has not experienced congestion. If congestion is experienced in any of the network elements along the VP/VC to the ATM destination terminal, the PTI field in the cell header is modified. The PTI field is therefore used for forward conveyance of the encountered congestion condition along the VP/VC to the ATM destination terminal.

The CLP field indicates the cell priority. This provides a capability for selective discarding under congestion conditions and therefore enabling network resiliency to traffic uncertainties. If the CLP is set to 1, it signifies that

the cell may be discarded in any of the network elements along the VC/VP path if local congestion above a threshold is encountered. The CLP indicator serves a dual purpose: setting of the CLP indicator in a cell to 1 by the sending terminal signifies that the cell carries non-essential information (and that the cell may be discarded under congestion); setting of the CLP indicator of a cell to 1 at the access point to the network indicates that the cell is in violation of the traffic limits agreed upon in the service contract. Setting CLP to 1 by the network for excessive traffic cells is termed a "traffic violation tag" for descriptive reasons: however, once the CLP has been set to 1, the handling of the cell is independent of the cause of this CLP setting [18]. The HEC field provides single bit error correction or multiple bit error detection capabilities on the cell header. The HEC monitors errors for the entire header.

The cell loss priority (CLP) and the payload type (PTI) are the two fields that enable the LW to provide the functionality discussed in this chapter.

3.1.2 Principles of operation

To illustrate the scheme let us consider Figures 3.2 and 3.3 which depict the PTI field of an ATM cell that is transmitted from one end-user to another over a network. End-users are shown as square blocks while routers are shown as circles. A user queues cells at the router. The router transmits cells which are

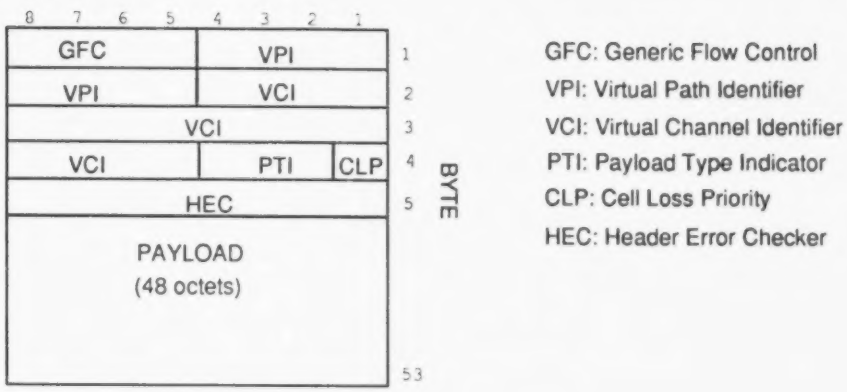

Figure 3.1: ATM cell structure

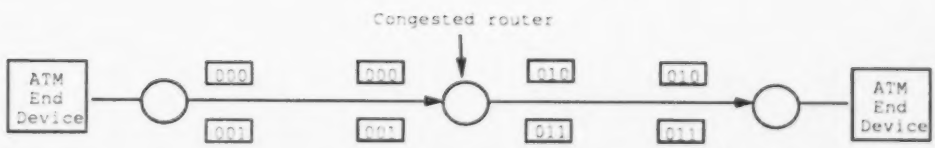

Figure 3.2: PTI field modified when congestion is encountered

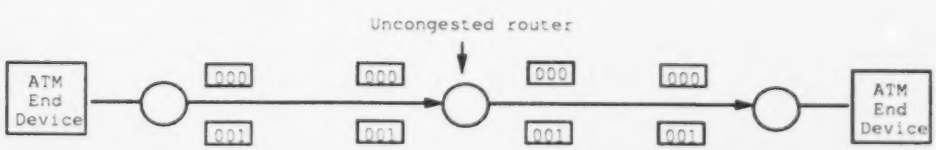

Figure 3.3: PTI field remains unchanged when no congestion is encountered

in the ATM cell format on the desired VC/VP. A cell may traverse multiple hops, one or more of which may be congested. When a cell encounters an intermediate router that is congested above a certain defined threshold, the router sets the second bit of the PTI field to 1. Thus for a cell that has not experienced congestion, the value of the PTI field which is either 000 or 001 is set to 010 or 011, respectively. We will refer to a network element that is congested above some defined threshold value simply as congested. A cell traversing a number of hops with an intermediate congested router is shown in Figure 3.2. The PTI field is originally cleared by the user to indicate no congestion. All routers that are not congested ignore the PTI field. Figure 3.3 depicts the situation where none of the routers is congested, and in this case, the PTI field remains unchanged, indicating no congestion.

Congestion at a router may be measured by the utilization of the router, the instantaneous queue length, or an average value of the queue length over an interval of time. The interval over which the average is performed will influence the responsiveness and stability of the scheme. In [48] an adaptive interval is recommended. This averaging technique will be discussed in more detail in a later section.

At the destination, cells belonging to a frame are accumulated until the entire frame is received. The number of cells that make up a frame may vary, so when a frame has been received, the PTI field of the last cell or an average of the congestion indications of all the cells of the frame received

may be used to determine the congestion feedback to be transmitted back to the source. This congestion feedback information is copied into the header of the acknowledgment frame. The acknowledgment frame is then transmitted by the destination to the source.

Once the acknowledgment frame is received, the user copies the congestion indication bit in the acknowledgment frame header into an appropriate data structure to be used by the window adjustment and estimation algorithm. Since the user clears the congestion indication bit before transmitting a cell, the congestion indication bit received in the acknowledgment frame indicates the state of the routers in the network. Users are required to adjust the traffic that they transmit based on their interpretation of the feedback from the routers in the network. This is achieved by users adjusting their window sizes. We define a window to be the value of the current window size and *credits* as the number of frames that may be transmitted by a user before the number of outstanding frames exceeds the window size.

In the LW scheme, when the assigned credits are exhausted, a user does not have to suspend its transmissions. Additional frames may be transmitted depending on an estimation of the availability of resources. The availability of resources is estimated by a user based on:

- The current window size, and
- The number of positive acknowledgments received within a given time

period.

These two measures indicate the rate at which frames are being transmitted by the router and the rate at which acknowledgments are being received from the destination. That is, conditions in the two way route from source to destination are reflected in negative acknowledgments, because on congested paths, marked cells may be discarded resulting in reception of incomplete frames at the destination, which subsequently are negatively acknowledged.

Using the window size and the number of positive acknowledgments received within an estimation interval, an estimate of the additional transmission capacity that may be used to transmit cells in excess of the window size is made. We will refer to the transmission capacity in excess of the window size as the *excess* capacity and the cells that are transmitted by a user using this capacity as *excess* cells. Cells that are transmitted within the window are not marked, and such cells are not discarded at routers if congestion develops. The performance of the network seen by users that are transmitting within the window size should not be affected by users transmitting excess cells. Thus, users may transmit excess cells on condition that they may be dropped in the event of congestion at any router in the network. Excess cells are marked by the user before transmission. A marked cell may be discarded at any router which is congested. At the destination, discarded cells will result in an incomplete frame or frames. Error detection procedures at the higher layers such as the adaptation layer or the transport layer flags such

frames. Entire frames that are discarded are detected by missing sequence numbers. In the LW scheme, frames that are received in error are selectively retransmitted. The choice of selective retransmission is based on studies in [45, pages 306–312] that has demonstrated its suitability for high speed networks. The operation of the LW is therefore similar to that of any sliding window mechanism with explicit congestion notification such as the explicit binary feedback scheme [48], except for the excess cells that may be transmitted by users. Transmission of excess cells in the LW has the following advantages over the conventional sliding window mechanism.

1. Transmission of excess cells will normally occur when a user has exhausted the assigned credits but knows that the network has available resources to handle a few more cells. Cells transmitted using excess bandwidth reduces the users backlog.

2. Acknowledgments received in regard to excess cells increase the rate at which the window adjustment is done in comparison to the conventional window mechanism. The LW is therefore, able to respond to changes faster than a comparable sliding window mechanism.

3. Real time traffic which requires a bounded delay are transmitted using excess capacity with some probability of success in this scheme. Using the window mechanism, this type of traffic may suffer excessive losses if the delay incurred while waiting for credits to become available exceeds

the permitted end-to-end delay.

The cost of permitting the transmission of excess cells is increased delay for frames whose cells may be discarded at congested nodes because such frames have to be retransmitted.

Routers and users are both responsible for implementing flow and congestion control in the network. The operations performed by the routers and users is outlined below.

3.1.3 Router operation

A router has two main processes, "receive" and "transmit" processes whose states are shown in Figure 3.4. These two processes may be executed concurrently when the buffer is not empty since the links are unidirectional.

The receive state

When a cell arrives at the router, it is queued for transmission if the buffer threshold B_T has not been exceeded. Marked cells are discarded on arrival at the router if the number of cells queued for transmission at the router exceeds B_T. All cells that are not marked are queued for transmission unless the buffer is full. If the buffer is full on arrival of a cell that is not marked, the arriving cell will be discarded only if there is no marked cell in the buffer,

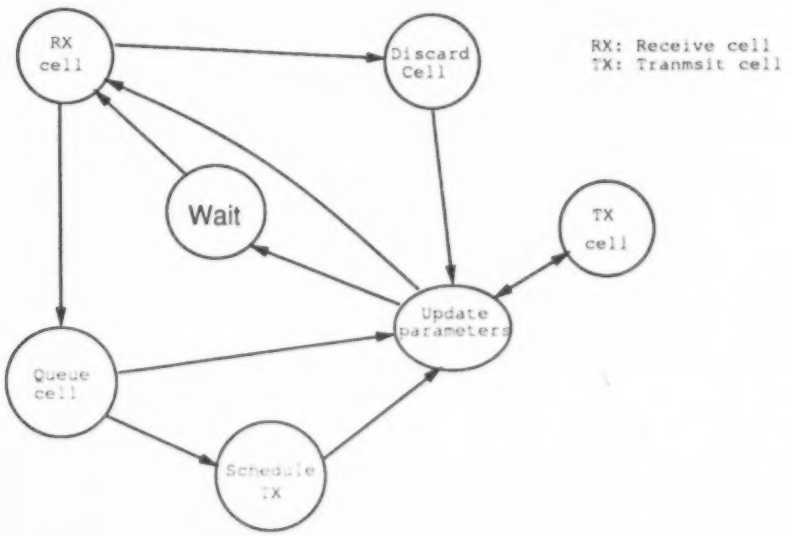

Figure 3.4: States of router processes

otherwise, it will replace the marked cell. The router queue length and cell loss record are updated every time a cell is received. If the buffer is empty at the arrival of a cell, the receiver schedules the transmission of the received cell in the next time slot. The receiver may go into the wait state if there are no incoming cells.

Transmit state

When the buffer is not empty, one cell is transmitted per time slot. After transmission of a cell, the queue length at the router is updated. The average queue length l of cells awaiting service is determined based on the number of cells queued for service at the router over an averaging interval T_{avg}. We

shall defer discussing the choice of T_{avg} to a later section. The transmitter may go into the wait state only when the buffer is empty. The instant when the transmitter resumes transmission after an idle period because the buffer was empty is referred to as the time of regeneration. T_{avg} is updated every regeneration, that is, when the transmitter resumes transmission after it was in the wait state.

Congestion detection

The router sets PTI = 000 and PTI = 001 to PTI = 010 and PTI = 011 respectively, in the cell header of cells arriving at the router when the average queue length, l, exceeds the threshold B_T.

Congestion control.

A marked cell is dropped if it arrives at a router when the buffer or queue length threshold B_T has been exceeded. Otherwise, it is queued for transmission. Cells that are not marked are all queued for service, except when the router buffer is full. In this case, all incoming cells are dropped. We note that the cell loss priority (CLP) bit in the ATM header can be used to implement this special priority scheme.

Figure 3.5: States of user processes

3.1.4 User operation

The user has two processes, "transmit" and "receive" processes. The states of the two processes are shown in Figure 3.5. In a similar way as the router processes, the user transmit and receive processes may executed concurrently.

Transmit frame

Traffic sources generate traffic according to the message arrival process. Messages generated for transmission are comprised of packets. A header and

trailer are added to form frames by the user. The user subdivides frames into cells and attaches the required header field before transmitting it to the router, such frames remain queued until a NAK or PAK is received. Before transmission, the PTI field of each cell is set to 000 except for the last cell of a frame in which the PTI field is set to 001 to mark the end of a frame.

Cells are transmitted from the user to the router at the access link speed, subject to availability of credits. If the credits are exhausted but excess capacity is available, the user marks cells before transmitting them to the router. When credits as well as excess capacity are exhausted, all arriving frames (newly generated as well as retransmission requests) are queued in the access queue until more credits become available. After transmitting a frame to the router, the number of credits assigned to the user is decremented by one, *i.e* one credit permits transmission of one frame.

Receive acknowledgment

Acknowledgments received may be positive (PAK) or negative (NAK). When a PAK is received, the number of credits is incremented, and the acknowledged frame is removed from the queue. Similarly, when a NAK is received, the number of credits is incremented but this time, the acknowledged frame is queued for retransmission. This is the "queue retx" state shown in Figure 3.5.

Parameter update

The window size is adjusted after receiving a number of acknowledgments equal to $2W$, where W is the current window size. Of the $2W$ acknowledgments received, only the last W are considered for purposes of window adjustment, while the rest are ignored. This is to take the effect of the last window adjustment into consideration. If more than 50% of the W acknowledgments considered for window adjustment carry positive feedback, the window size is incremented. Otherwise, it is decreased geometrically by multiplying it by a reduction factor. In addition, an estimate of the excess capacity is made every 2τ, where τ is the one way propagation delay.

3.1.5 Receive cell

The destination receives cells and acknowledges all received frames. A user enters the receive state depicted in Figure 3.6 due to the arrival of a cell. In this state, the user performs the following operations. Arriving cells are accumulated ("queue cell" state, shown in Figure 3.6), to reconstruct frames before passing them on to higher layers. If, after a cell is queued, the end of frame is detected, then an acknowledgment for the received frame is sent. Otherwise, the receiver stays in the wait state. The received frames can be regarded as either complete or incomplete. Complete frames are frames with no lost cells, while any frame with one or more lost cells is considered

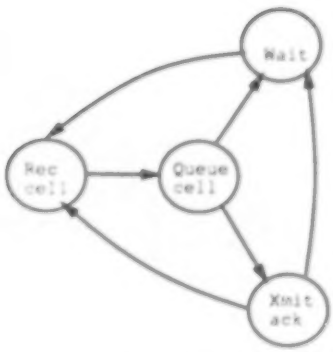

Figure 3.6: User receiver state diagram

incomplete. Incomplete frames and frames received with errors are detected at the ATM adaptation layer through the use of:

1. The frame length which is transmitted as part of the frame,
2. Error detection of received frame by cyclic redundancy checking (CRC), and
3. A reconstruction timer which limits the maximum duration of time required to reconstructed a frame.

The destination therefore accumulates cells until an entire frame or an incomplete frame is detected. For each received frame, an acknowledgment is sent by the destination. The acknowledgment is positive for complete frames while it is negative for incomplete ones.

When the received frame is complete and is in sequence, it is positively acknowledged, and the user either returns to the wait state or continues

to receive incoming frames. If the frame is incomplete but in sequence, a negative acknowledgment is sent to the source as mentioned above, and serves as a retransmission request. Such incomplete frames are discarded by the destination. If the received frame is out of sequence (complete or incomplete), then it is either a retransmitted frame or there are missing frames between the received frame and the expected one. A retransmission request is made for all missing frames.

Acknowledgment frames are also used to deliver feedback of congestion experienced by the cells of the frame being acknowledged to the source. The congestion status is determined for the PTI field of each cell as follows. The congestion sign in the PTI field in each received cell is noted. The congestion feedback sign sent to the source is based on a majority of the congestion sign in the PTI fields of cells in a frame. The feedback is copied into the header of the acknowledgment frame, which is transmitted to the source of the frame.

In the remainder of this chapter, the router and user operations are presented in detail.

3.2 Router Operation

In this section, we discuss the policy of congestion feedback signal generation due to congestion at routers feedback filtering and congestion control.

3.2.1 Feedback signal generation

In the LW scheme, each router is viewed as a single server. The service discipline of the server may be first-come-first-served (FCFS) or round robin [29]. Different sources of traffic may share the same path, or have different paths, only interacting at routers common to the different paths. A common buffer is shared by all the traffic that passes through a router. If congested, a router sets the PTI = 000 and PTI = 001 to PTI = 010 and PTI = 011 respectively, of each arriving cell to indicate a state of congestion at the router. Congestion at a router may be detected by the level of utilization of the router, by queue length or both.

The utilization of a router depends on the service requirements of each virtual channel with the router in its path. A router serving virtual channels with different paths may be heavily utilized depending on the service requirements of each of the virtual circuits. Utilization of a router therefore, may vary considerably depending on the traffic in the paths of virtual channels that are served by the router. With such a variation, utilization is not a good indication of path congestion. Therefore, in this scheme, congestion detection is based on the queue length at the router. A single threshold value, B_T, is used as a decision point for feedback generation. This is chosen based on studies in [48], which test the policy of using a single threshold as well as hysteresis to set the congestion indication in the PTI field in the header of

a cell. In [48], it is found that the *power*[1] is maximized when hysteresis is nonexistent with a threshold value = 1.

3.2.2 Feedback filtering

The congestion indication bit in a cell header is set if the average queue length at the router exceeds the threshold B_T. As mentioned earlier, congestion detection at a router is based on the average queue length rather than on the instantaneous queue length, since using the instantaneous queue length to indicate congestion may trigger false alarms causing users to reduce their window sizes while the router is not necessarily congested. This may also result in an unacceptably low throughput.

The queue length is therefore averaged over a certain period of time. A number of averaging methods have been proposed but we choose to use the adaptive averaging in this scheme. In the adaptive averaging method, the cycle time is determined by a router adaptively. A cycle, T_c, as shown in Figure 3.7, is defined as a *busy + idle* interval seen at the router. This interval is also called the 'regeneration cycle' and the beginning of the busy period is called a regeneration point. The average queue length is given by the area of the shaded curve shown in Figure 3.7 divided by the cycle time. This average will be used for the duration of the next cycle. Some

[1] $\alpha = 1$.

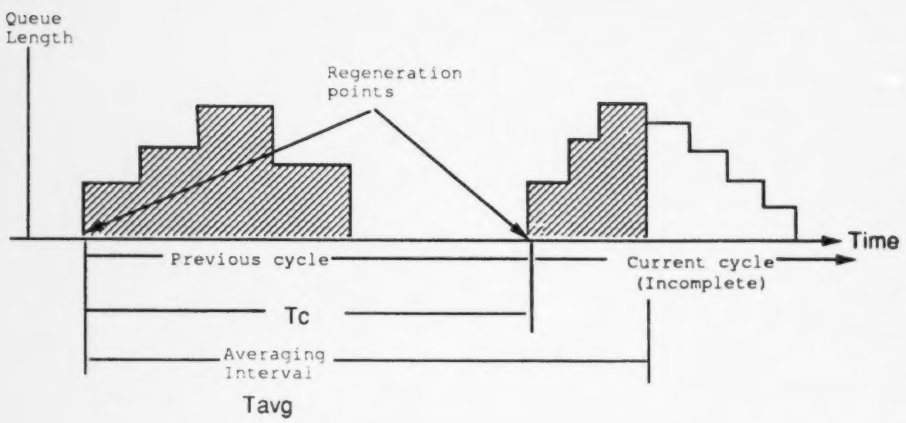

Figure 3.7: Adaptive averaging

refinements to account for the case where the regeneration cycle may be very long is adopted. For example, when the busy period is very long, it may be necessary to reflect a more current average queue length than the previous cycle's average. The feedback based on the previous cycle may not reflect the current situation. Therefore, basing the average queue length on the previous cycle as well as the current, though incomplete cycle, as shown in Figure 3.7 will take the condition of both cycles into account. The queue length average l is calculated by computing "the area under the queue length curve" since the beginning of the "previous" cycle and dividing it by T_{avg}. The averaging is therefore performed as each cell arrives at the router. Thus, as the current cycle gets longer, the average due to the current cycle begins to dominate and the average of the previous cycle decays.

3.2.3 Congestion control

Marked cells may be discarded depending on the state of a router when a cell arrives at it. If the instantaneous queue length L is less than the threshold B_T, the cell is queued for transmission at the router. Marked cells are served in a way similar to other cells of the same connection that are not marked. If the service discipline at a router is prioritized, all cells of a connection are served according to the priority of that particular connection. If the number of cells queued at a router exceeds B_T, marked cells are discarded. When the buffer is full, unmarked arriving cells will be discarded only if there are no marked cells in the buffer. Otherwise, a marked cell in the buffer is discarded in order to accommodate the arriving cell that is unmarked.

3.3 User operation

A number of decision parameters may affect the window adjustment policy:

- The choice of the frequency of window size adjustment
- The choice of the type of congestion feedback information
- Use of congestion feedback information (signal filtering)
- The window size increment and decrement algorithms
- The excess capacity estimation algorithms

- The maximum time that an established connection may be idle for and, on resumption of transmission, continue using a window size equal to the size at the start of the idle period.

- The initial window size of a user that has an established connection when it resumes transmission, after it has been idle long enough to be regarded as a dormant user

- Re-adjustment of the window size by users that have been transmitting using a capacity less than the current window size for a period of time that exceeds some preset limit, to the size required for the current rate of traffic generation. Such a situation may occur when an active user at some point has less traffic to transmit than permitted by the current window size.

3.3.1 Window size adjustment frequency

First, we define W_{i-1} and W_i to be the window size before adjustment and the window size after adjustment, respectively. We distinguish W_{i-1} from W_i for clarity of exposition and to maintain consistency with the conventional window mechanism. We shall justify why it is not necessary to distinguish between W_{i-1} and W_i, thus we use a single value of the window size, W, equal to W_i in the window adjustment and estimation algorithm. It is assumed that the user has sufficient traffic to fill the window size in following discussion.

In a window mechanism, window size adjustment is done after receiving $W_{i-1}+W_i$ acknowledgments since the window was last adjusted. The window size is adjusted after receiving $W_{i-1}+W_i$ in order to take into consideration the effect of the previous window size adjustment, before a new adjustment is made. Suppose the window size adjustment policy increases the window size additively by w and decreases the window size multiplicatively by a factor r_{fact}. In Figure 3.8, we denote the one-way propagation delay by T. As shown in Figure 3.8, the window size is increased by w at time t_1, to make $W_i = W_{i-1} + w$. At time t_1, there are W_{i-1} outstanding frames. The window size will be adjusted again after receiving $W_{i-1} + W_i$ acknowledgments. Congestion feedback from the first W_{i-1} received acknowledgments will be ignored since these acknowledgments are due to frames transmitted before the adjustment of the window size. $W_{i-1} + w$ acknowledgments are considered in increasing the window size which, as shown, in Figure 3.8 will happen at time $t = t_1 + 4\tau$. Clearly, this corresponds to adjusting the window size after approximately 4τ, where 2τ is the round trip propagation delay, given that the window size, W, is small compared to 2τ.

In the LW, we maintain the window adjustment frequency to be every $W_{i-1} + W_i$ of received acknowledgments. We define W_e to be the excess capacity estimated by a user in the LW mechanism. We shall refer to the excess capacity, W_e, as excess credits since with this excess capacity, W_e frames may be transmitted. W_e excess credits in addition to W_i credits are

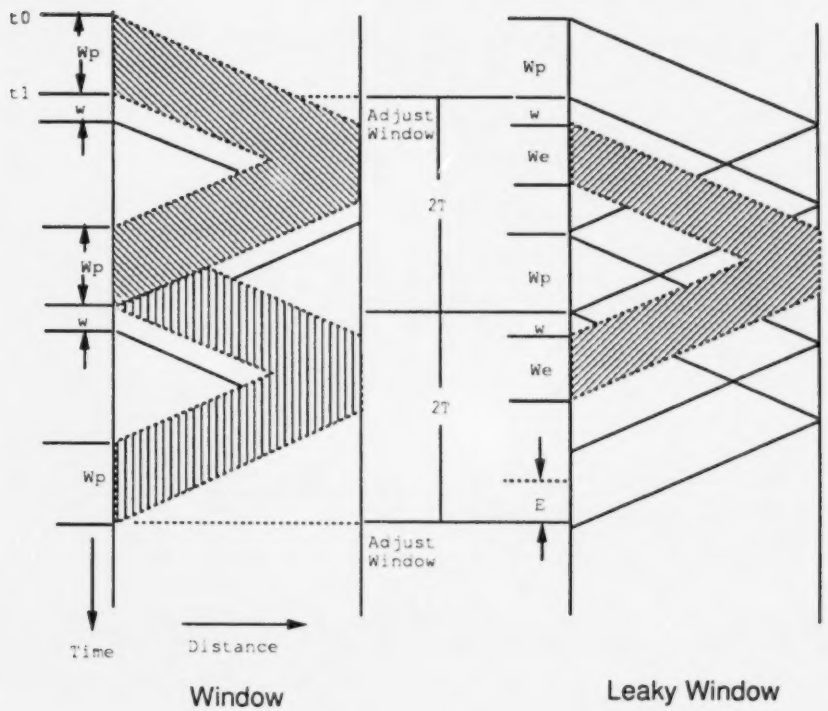

Figure 3.8: Window adjustment space time diagram.

therefore available to a user in the LW mechanism. Similar to the window mechanism, the current window size $W_i = W_{i-1} + w$ after it is incremented. The shaded region of the LW diagram in Figure 3.8 is the excess window size W_e. As stated earlier, credits are in terms of frames, but transmission of frames is in the form of cells.

With W_e excess credits, W_e excess frames may be transmitted. We define E to be the number of received acknowledgments due to the transmission of excess cells in addition to $W_{i-1} + W_i$ that would be received when using the window mechanism in a window adjustment interval. Two methods of frame transmission are proposed. The first method does not distinguish between the two types of credits which means that any mixture of marked and unmarked cells may be transmitted within a frame. The second method distinguishes excess credits from normal window credits. We shall discuss the implications of the use of each of the two methods.

1. Using the first method of frame transmission, a mixture of marked and unmarked cells within a frame may occur as follows. Suppose that during transmission of some cells, only excess credits are available. The cells transmitted will therefore be marked. Furthermore, suppose during the transmission of marked cells and before the entire frame has been transmitted, additional credits become available. Cells transmitted after the arrival of credits will not be marked. Clearly, a mixture of marked and unmarked cells can be transmitted within the same frame.

The decision to decrement either the credits or excess credits after a frame has been transmitted is made as follows: The number of credits is decremented if the cells transmitted in a frame are all unmarked or partly marked, while the number of excess credits will be decreased only when all the cells of a frame transmitted are marked. The significance of transmitting marked cells in this case is that, in total, we expect a higher rate of transmission than in the window mechanism. This is reflected in the number of received acknowledgments.

As has already been stated, the window size is adjusted after receiving $W_{i-1} + W_i$ acknowledgments. Some of these acknowledgments are due to marked cells transmitted while others are due to transmission of unmarked cells. In order to evaluate the effects of transmitting marked cells on the rate of window adjustment, we should identify acknowledgments received due to the transmission of marked cells. This, however, is not possible since acknowledgments do not distinguish between the use of credits or excess credits. For discussion purposes, we let E be the number of received acknowledgments due to the transmission of marked cells within a window adjustment interval. Therefore, in one window adjustment interval, we can say that E of $W_{i-1} + W_i$ received acknowledgments are due to the transmission of marked cells.

2. Using the second method, all cells in a frame are transmitted either

marked or unmarked. After a frame is transmitted, the decision to decrement either credits or excess credits is straightforward because if all the cells transmitted are marked, then the number of excess credits is decremented; otherwise, if all the cells transmitted are unmarked, then the number of credits is decremented. In this case, E, which is defined as the the number of acknowledgments received due to the transmission of excess cells in a window adjustment interval has a definite value. However, the actual value may not be known, because entire frames may be discarded, which means acknowledgments for such a frame will be received after the destination detects that there is a missing sequence number. Some acknowledgments may therefore arrive after the end of window adjustment interval in which it was expected.

Using either of the frame transmission methods discussed above, E additional acknowledgments will be received within 4τ. The expected window size adjustment interval in the LW is therefore reduced to $4\tau - E$.

Finally, let us consider the effects of resetting the window size by a user after it has been idle for a predefined time duration. Resetting the window size does not affect the basic interval of window adjustment because users are allowed an idle period greater than 4τ, which is greater than the adjustment interval, before any reduction in their window sizes can be implemented. This interval is chosen in order not to unnecessarily throttle users that may be waiting for outstanding frames to be acknowledged.

3.3.2 Signal filtering

The decision to increase the window size is made by the user based on a function of the number of positive congestion feedback received within W acknowledgments. We use the threshold function determined in [48] by which the window size should be increased if the positive feedback exceeds 50% of W acknowledgments received, or is reduced otherwise.

3.3.3 Window size adjustment and estimation of excess capacity

An additive increase/multiplicative decrease algorithm is used to adjust the window size W. The additive factor is 1, while the multiplicative reduction factor is a parameter r_{fact}. After adjusting the window size, W_i be greater than W_{i-1} when the window size is increased. But when the window size is decreased, W_i is less than W_{i-1}. However, in the window adjustment and estimation algorithm, we let $W = W_i$ for the following reasons.

1. Increasing the window size by one does not introduce a significant difference between W_{i-1} and W_i.

2. When the window size is reduced by a factor, $r_{fact} < 1$, the current window size W_i becomes smaller than W_{i-1}. Depending on the value of r_{fact}, W_i may become significantly smaller than W_{i-1}. In

the window mechanism, window adjustment is done after receiving $W_{i-1} + W_i$ acknowledgments. If no distinction between W_i and W_{i-1} is made, *i.e.* $W = W_i$, then the window is adjusted after receiving $2W = 2r_{fact} \times W_{i-1}$ acknowledgments, which may be less than $W_i + W_{i-1}$. Thus after the window size is decreased, the rate of window size adjustment will be higher than in the case where explicit values of W_{i-1} and W_i are used. A higher rate of window adjustment is desirable as it enables users to respond to the state of the network faster.

A similar window adjustment policy has been proposed in [2].

The excess capacity that a user may use to transmit marked cells is based on an estimate of the network load along the path of the connection under consideration. The estimation is performed as follows: each user is assumed to have an estimate of the round trip delay $2\tau_i$ of connection i. The estimated excess window size, W_e^i, of a connection i is calculated as

$$W_e^i = A_{pos}^i(2\tau) - \epsilon_{fact}^i \times W^i,$$

where $A_{pos}^i(2\tau)$ is the number of positive acknowledgments received within $2\tau_i$, W^i is the window size and ϵ_{fact}^i is an estimation factor. The values of ϵ_{fact}^i are determined experimentally.

The rationale behind this estimation technique is as follows. In Figure 3.8, we show that one expects to receive W^i acknowledgments within a round trip time interval $2\tau_i$ if the network is uncongested. We permit some excess cells

to be transmitted and measure $A^i_{pos}(2\tau)$, the actual number of positive acknowledgments received in $2\tau_i$. The number of positive acknowledgments received in any time interval indicates the resources available to the connection during the transmission time, which is equal to $A^i_{pos}(2\tau)$ frames over a time interval of $2\tau_i$ for user i. Therefore, if $A^i_{pos}(2\tau) - W^i > 0$, then user i could have used $A^i_{pos}(2\tau) - W^i$ additional credits without causing congestion in the network. We make a prediction of the excess capacity for user i based on this quantity by introducing an estimation factor ϵ^i_{fact}.

In making the estimate W^i_e, we attempt to get an idea of the number of additional frames that may be transmitted by a user without causing congestion in the path of the connection. An interval that is a multiple of $2\tau_i$, the round trip propagation delay, is chosen because over such an interval, a user that has sufficient load is expected to receive acknowledgments equal to the window size W^i. The value of $2\tau_i$ need not be known precisely, but it should be kept constant for the duration of the connection.

An aspect that came out of experiments is that we can get a lower probability of loss by increasing the frequency of estimation. For example, a user may observe the number of PAK received in τ_i instead of $2\tau_i$. The PAK received in τ_i is modified to reflect the number of PAK that would be observed in $2\tau_i$. This is done by introducing a gain factor G into our estimation equation. G adjusts the number of PAK received in τ_i to reflect the number of PAK that would be received if an observation of acknowledgments was made

over an interval of $2\tau_i$. The estimation equation can thus be rewritten as

$$W_e^i = G \times A_{pos}^i(\tau) - e_{fact}^i \times W^i$$

3.3.4 Duration of the idle period

An established connection may become idle temporarily, but the users may wish to maintain the connection. When such users restart transmission, the window size and the estimate of the excess window before the connection became idle need to be modified to reflect the current state of the network. This is due to the fact that changes in the network are fast and the load in the network may have changed considerably depending on how long a connection has been idle. We introduce a limit T_{idle}, of how long a connection that is temporarily idle is considered to be still active. T_{idle} is set to $n \times \tau$, where n is a factor determined experimentally. A user that becomes temporarily idle may resume transmission after the idle period with the window size unadjusted if the duration of the idle period is less than T_{idle}. When the idle period is greater than T_{idle}, the user is required to resume transmission with a window size equal to the initially negotiated size, W_{in}. The user also resets the estimate, $W_e = 0$.

Users that have window sizes in excess of their requirement will be able to transmit all their traffic within the assigned capacity. If the duration of congestion at intermediate nodes or routers is short, then a sufficient number

of cells from such users may pass the points of congestion without having their congestion notification bit set to 1 for two reasons:

1. The interval of transmission of successive frames by such users may be long compared to the duration of congestion at intermediate nodes.

2. Since there is usually no shortage of credits, a frame is transmitted as soon as it arrives at its source. Therefore, frames are equally spaced apart, in time. Since congestion is usually short-lived, very few frames may encounter it. Therefore, the number of negative acknowledgments received is insufficient to cause such users to reduce their window sizes.

If such a situation occurs, the users will receive acknowledgments, a majority of which will carry positive feedback. This may result in a monotonic increase in the window size of some users while for others the window sizes may fluctuate. Such an occurrence is undesirable because

1. Occasionally, users with large window sizes may have sufficient traffic to fill their window size, causing a huge burst of traffic in the network. This may result in big losses.

2. The LW permits transmission of marked cells on condition that they may be discarded at congested nodes. Users with disproportionately longer windows will be allowed to transmit a big percentage their traffic unmarked, which may result in an increased number of marked cells

being discarded. This may be considered an unfair allocation of available transmission capacity and may also degrade the performance of the LW mechanism.

Based on these arguments, a user whose window size is in excess of its requirement will have to reset its window size to its required capacity after a time T_{reset}, to prevent a monotonic increase in the window size.

3.4 Summary

In this chapter, the LW mechanism was proposed and its principle operations discussed in detail. This mechanism should be implemented at the ATM adaptation layer. The CLP and the PTI fields in the ATM cell header enable the LW to provide the functionality discussed in this chapter.

The LW mechanism uses the same principles as the window mechanism with the modification that it permits cells in excess of the window size to be transmitted by users. The number of excess cells that a user may transmit at any given time is based on an estimate of the network load.

Excess cells are transmitted on condition that they may be discarded at any congested node. The user is therefore required to distinguish excess cells by setting the CLP to 1 for excess cell and to zero otherwise. On arrival at a congested router, a cell with the CLP set to 1 may be discarded.

The network load is measured by routers using an average length of the

cell queue. The averaging interval is determined adaptively by each router. Whenever the average queue length exceeds a set threshold, the PTI field of arriving cells is modified to indicate congestion experienced. In addition, the router discards all cells which have the CLP bit set to 1 when the instantaneous queue length exceeds the threshold.

At the destination, cells are accumulated to reconstruct frames from the received cells. Complete frames that are error free are positively acknowledged, while incomplete frames or frames received in error are negatively acknowledged. The destination determines the sign of the congestion feedback to be sent to the source by a majority sign of the congestion indications in the PTI fields of cells in the frame.

Users periodically adjust their window sizes based on the feedback in the received acknowledgment. In addition, an estimate of the excess capacity is made every 2τ.

Chapter 4

Simulation model and results

In this chapter, the implementation of the LW mechanism and the network architecture used to test it are studied using a simulation model. The simulation model implements both the user and the router functions. A description of the implementation window mechanism and the virtual leaky bucket. Finally, results of performance of the LW and comparisons to other schemes are presented.

4.1 Description of the model

In this simulation model, a two phase traffic generator with geometrically distributed active and silence periods is used for traffic generation by each user. First, we shall define the terms used in this chapter.

4.1.1 Definition of terms used

message_queue: queue of messages with frames that have not been acknowledged

access_queue: pointer to the next frame awaiting transmission in the message_queue

retransmit_queue: queue of frame sequence numbers to be retransmitted

current_time: the clock or global value of time

W: window size

W_{max}: maximum permitted window size

W_{min}: minimum negotiated window size

W_e: excess window size

W_u: number of frames transmitted which have not been acknowledged (outstanding frames)

W_m: number of outstanding frames that are marked

W_{in}: initial window size of the connection, which may be different from the minimum window size

T_{idle}: the maximum duration of time an established connection may be idle and on resumption of transmission may continue using W, the window size at the start of the idle period

T_{reset}: the maximum duration of time an active user may transmit at a rate less than that permitted by the current window size without being required to readjust its window size to its current rate of transmission

A_{pos}: number of positive frame acknowledgments received by a user in an estimation interval

fback: number of positive congestion feedback indications received in a window adjustment interval

b: number of router cell buffers occupied at any time instant

B_T: buffer threshold

e_{fact}: estimation factor

r_{fact}: window reduction factor

β^{-1}: mean of duration of the traffic generator in the active state

α^{-1}: mean duration of the traffic generator in the silent state

γ^{-1}: mean inter-arrival time of frames in active state

When each of these terms refers to an individual user, it is to be understood that it will be implicitly indexed by the user number.

4.1.2 Message generation

The active and silent states of the two phase traffic generator are geometrically distributed with means β^{-1} and α^{-1}, respectively. A single message consisting of a number of frames is generated in one active period. In the active state, frames are generated according to a geometrically distributed inter-arrival time t_i, with mean γ^{-1}. The frame length is chosen to be either 2 or 20 cells with equal probability. This is the nature of traffic in diskless work stations on an Ethernet [27]. After initialization, the traffic generator schedules an active period for each user i to begin at a randomly chosen time, t_g. A geometrically distributed active period, lasts for t_a time units. Frames of length 2 or 20 cells are generated during the active period and queued at the users **message_queue**. At the end of the active period, a geometrically distributed silence period t_s starts. The next active period is scheduled to start at the end of the silence period. Traffic generation is summarized by the pseudocode in Figure 4.1 which uses a random number, u, between 0 and 1. Each frame is assigned a sequence number upon generation. The sequence numbers used are in the range of 0 to 2^{31}.

```
Procedure: Generate message
        time := t_g                                          Initialize variables
        message_queue := NULL
        access_queue := NULL
        retransmit_queue := NULL
        begin
           repeat
              u := random()
              current_time := time
              t_a := ⌈ln(u)/ln(1-β)⌉
              while (current_time < time + t_a) {            Active period
                    u := random()
                    if (u > 0.5)
                       frame_size := 20
                    else
                       frame_size := 2
                    queue_frame(message_queue)
                    u := random()
                    t_i := ⌈ln(u)/ln(1-γ)⌉
                    sleep(t_i)
                    current_time := current_time + t_i
                    }
              u := random()
              t_s := ⌈ln(u)/ln(1-α)⌉
              sleep (t_s)                                    Silent period
              time := current_time + t_s
           until(doomsday)
        end
```

Figure 4.1: Message generation

4.1.3 Frame transmission

We will assume that each user has a transmission buffer which stores the frame being transmitted. The transmission buffer is assumed to be of sufficient size to accommodate a frame. The user partitions a frame and adds the required header to form ATM cells before the start of transmission. As depicted in the state diagram in Figure 3.5 and, as already discussed in chapter 3, a user may start transmission when one of the following events occur:

- When a new frame is received from the message generator, and transmission credits are available,
- When a retransmission request for a frame is received, or
- When additional unused capacity becomes available due to the arrival of an acknowledgment which increases the number of available credits or additional excess capacity.

There are two modes of transmission. The first, which will be referred to as strategy A, allows for transmission of a mixture of marked and unmarked cells within a frame. The second method distinguishes between credits and excess credits. All cells within a frame are transmitted using either credits or excess credits. We shall refer to this method as strategy B. The transmission algorithm for the strategies A and B are given in Figures 4.2 and 4.3, respectively. The two algorithms are the same except for the control

of cell transmission. Before describing these algorithms, the following additional terms are defined first. The procedure *fetch_frame* acquires a frame from a specific queue into the transmission buffer and advances the pointer (access_queue or retransmit_queue) to the next frame awaiting transmission. The procedure *transmit_cell* queues one cell per time slot at the transmission link. *xmit_count* is the number of cells in the transmission buffer. The flag *restricted* is used in strategy B to indicate that transmission using excess capacity (marked cells) is in progress. The flag is reset after an entire frame has been transmitted.

Retransmission requests are given a higher priority than newly generated frames. When fetching a frame to be transmitted, the retransmission queue is inspected first. The transmission buffer is loaded with a frame requested for retransmission if the retransmit queue is not empty. When the retransmit_queue is empty, the frame pointed to by the access_queue is loaded into the transmission buffer. Transmission of a frame is subject to the availability of credits. Marked cells are transmitted in strategy A only when excess capacity is available and all the credits are exhausted. From the instant that credits become available until all credits are exhausted, all cells transmitted are not marked. In other words, transmission using excess capacity stops when credits become available and may resume only when credits are exhausted.

In strategy B however, arrival of additional credits does not necessarily pre-empt transmission using excess capacity. If on arrival of additional credits, transmission of a frame using excess capacity is in progress, then the remaining cells in the frame will all be transmitted using excess capacity. Thus all the cells in the frame are marked. Cells transmitted subsequent to the end of a frame whose cells are marked will be transmitted without marking until the credits are exhausted. In both transmission strategies, the PTI field in the cell header is set to 000 except for the last cell of the frame that this field is set to 001 to mark the end of frame.

When credits as well as excess credits are exhausted, retransmission requests or newly generated frames will wait in the retransmit or access queue respectively, until additional credits or excess credits becomes available. A user may become idle because there is nothing to transmit, which corresponds to the wait state in Figure 3.5. On resumption of transmission activity, if a user has been idle for a time period longer than T_{idle}, it resets the window size to W_{in}. As discussed in Chapter 3, an active user that has been transmitting at a rate less than that permitted by the current window size for longer than a preset time interval, T_{reset}, is required to reduce its window size. In the implementation, the window size is reduced to the maximum size used in the interval T_{reset}. Until the transmission buffer becomes empty, cells are transmitted to the router at the access line speed. We assume that the propagation delay from the user to the router is zero and the access line speed is

the same as the trunk line speed. Therefore one cell is queued at the router per time slot until the entire frame has been transmitted. The transmission buffer can be loaded with another frame when it becomes empty.

4.1.4 Receive cell

Cells received by the ATM layer at the destination are passed on to the adaptation layer for frame reconstruction and error detection before being passed on to higher layers and before an acknowledgment is sent to the source. Figure 4.4 is the algorithm executed by the cell receiver. In this algorithm, the operations performed to reconstruct frames from received cells, congestion feedback generation, request for retransmission of incomplete frames or frames received in error are included. The other operation belong to the adaptation layer and are outside the scope of this thesis. Frame sequence numbers are transmitted as part of a frame and are recovered after a frame is reconstructed.

The terms used in the algorithm will be defined first. *bits_set* counts the number of cells within a frame which have the PTI field indicating no congestion. *cell_count* is the number of cells of a frame that have so far been received. *cell_rec* is the total number of cells received at the destination for complete frames. *cell_rec* is used to calculate the average link throughput at the end of the simulation run. *frame_delay()* computes the average and the

```
Procedure: Transmit frame
        W := W_in
        W_e := W_u := W_m := 0
        begin
           repeat
              if (W_u > W and W_m > W_e)                    Wait, no credits
                wait_event(credits or excess_bandwidth)
              if (retransmit_queue ≠ NULL)
                fetch_frame(retransmit_queue, xmit_buffer)
              else if(access_queue ≠ NULL)
                fetch_frame(access_queue, xmit_buffer)
                else
                     wait_event(frame_arrival)
              xmit_count := frame_size
              while (xmit_count > 0) {                      Transmit frame, cell by cell
                   if(xmit_count = 1)                       Not end of frame
                     PTI := 000
                   else                                     End of frame
                     PTI := 001
                   if (W_u < W)                             Cell not marked
                     transmit_cell(xmit_buffer)
                     xmit_count := xmit_count - 1
                     if (xmit_count = 0)
                        W_u := W_u + 1
                   else if (W_m < W_e)                      Cell marked
                     mark_cell
                     transmit_cell(xmit_buffer)
                     xmit_count := xmit_count - 1
                     if (xmit_count = 0)
                         W_m := W_m + 1
                  }
           until(doomsday)
        end
```

Figure 4.2: Strategy A frame transmission algorithm

```
Procedure: Transmit frame
        W := W_in
        W_e := W_u := W_m := 0
        restricted := 0
        begin
           repeat
              if (W_u > W and W_m > W_e)                    Wait, no credits
                  wait_event(credits or excess_bandwidth)
              if (retransmit_queue ≠ NULL)
                fetch_frame(retransmit_queue, xmit_buffer)
              else if(access_queue ≠ NULL)
                fetch_frame(access_queue, xmit_buffer)
                else
                     wait_event(frame_arrival)
              xmit_count := frame_size
              while (xmit_count > 0) {          Transmit frame, cell by cell
                   if(xmit_count = 1)                    Not end of frame
                     PTI := 000
                   else                                      End of frame
                     PTI := 001
                   if (W_u < W and restricted = 0)           Cell not marked
                      transmit_cell(xmit_buffer)
                      xmit_count := xmit_count - 1
                      if (xmit_count = 0)
                         W_u := W_u + 1
                   else if (W_m < W_e)                       Cell marked
                      restricted := 1                        Set flag
                      mark_cell
                      transmit_cell(xmit_buffer)
                      xmit_count := xmit_count - 1
                      if (xmit_count = 0)
                         restricted := 0                     Reset flag
                         W_m := W_m + 1
                   }
           until(doomsday)
        end
```

Figure 4.3: Strategy B frame transmission algorithm

distribution of the end-to-end frame delay. *max_frame_size* is the maximum number of cells permitted in one frame. Max_frame_size detects loss the cell that marks the end of frame. Loss of part or an entire frame is detected by the reconstruction timer, which limits the maximum duration of time, T_1, that it may take to reconstruct a frame. The actual values of T_1 that may be used have not yet been standardized. The simulation model used does not implement the reconstruction timer, but instead uses frame sequence numbers to determine cell loss. The sequence number is not transmitted as part of a cell but it is included in the data structure of a cell. Cells are accumulated by the routine *queue_cell()*. The end of a frame is detected by the adaptation layer on arrival of a cell with the PTI field in the header set to either 001 or 011. The acknowledgment signs for complete and incomplete frames are positive and negative, respectively. The routine *make_ack()* makes acknowledgments based on the cell received in a frame. Congestion feedback carried by the acknowledgment is based on bits_set, the number of cells received which have the PTI field equal to 000 or 001. The routine *make_feedback()* determines the congestion feedback sign. Cells of an entire frame discarded at congested nodes will result in missing sequence numbers at the destination. We shall refer to such frames as missing frames. The order of sequence numbers of received frames is checked to determine whether there are any missing frames. Retransmission requests for the missing frames are submitted by the destination. The sequence numbers of the frames requested for retransmission by

the destination are queued in the *missing_frames* queue in the order of the expected arrivals of retransmissions. That is, the sequence number at the head of the missing_frames queue is expected to arrive first and the sequence number at the tail of the queue is expected last.

We assume that the reception of an unexpected retransmitted frame implies the loss of frames preceding the received frame in the missing_frames queue within the network. Therefore, when a retransmitted frame that is not at the head of the missing_frame queue is received, all sequence numbers in the missing_frames queue that precede the received frame are placed at the tail of the queue and another retransmission request for these frames is made. In the model, we assume that the loss of cells, and hence frames, in virtual channels occurs only because of discarding at congested routers. This justifies the assumption that frames that arrive out of the expected order are due to lost frames.

4.1.5 Receive acknowledgment

When an acknowledgment is received, the major operation performed is to update the window control parameters. The parameters include W, W_u, W_m, F_r, A_{pos}, and $fback$. The operations performed when an acknowledgment is received are shown in Figures 4.5 and 4.6. Window adjustments are done every $2W$. Feedback received in the last W of those $2W$ received acknowl-

```
Procedure: Receive cell
    cell_count := bits_set := 0                                  Initialize variables
    begin
       repeat
          wait_event(cell_arrival)
          queue_cell()
          cell_count := cell_count + 1
          if (cell_count + 1 < max_frame_size and      Is frame size > maximum
                                 timer < T1)           frame size or incomplete?
             if ( PTI = 000 or PTI = 001 )
                bits_set := bits_set + 1
             if (end of frame )              Adaptation layer checks end of frame
                if (frame error-free )                              and frame error
                   make_feedback(bit_set)
                   make_ack()
                   transmit(acknowledgment)
                   frame_delay()
                   cell_rec = cell_rec + cell_count
                   bits_set := cell_count := 0
                else                                                Error in frame
                   transmit(retransmission_request)
                   queue_seq_no(missing_sequence)
                   cell_count := bits_set := 0
                if (frame not in sequence)        Missing frame, request retransmit
                   if (retransmitted frame)
                      remove_seq_no(missing_sequence)          Remove from queue
                      if (frame not in sequence)
                         transmit(retransmission_request)
                         queue_seq_no(missing_sequence)
                   else
                      transmit(retransmission_request)
                      queue_seq_no(missing_sequence)
          else                                                Frame too long or incomplete
             transmit(retransmission_request)
             queue_seq_no(missing_sequence)
             cell_count := bits_set := 0
       until(doomsday)
    end
```

Figure 4.4: Receive cell and make ack when frame is complete

edgments are used to decide how to adjust the window size. *adjust_flag* identifies when to make the window adjustment. If the acknowledgment received is negative, the frame with the sequence number in the received acknowledgment is queued for retransmission.

We note that only feedback from positive acknowledgments is accumulated in *fback* for the purposes of window adjustment since negative acknowledgments are due to discarded cells at intermediate nodes resulting in incomplete frames being received at the destination. Feedback from such frames is therefore is ignored. This is equivalent to assuming the feedback from such frames is negative.

Every 2τ time units, the excess window size is estimated. Estimation is done by the function *estimate_excess* in Figure 4.6. This is the estimation formula discussed in Chapter 3. Finally, messages that have successfully been transmitted are removed from the message_queue at the source.

4.1.6 Router activity

The operations of the router will be presented in terms of the receiver and the transmitter. The operations of the receiver are given in Figure 4.7. Upon a cell's arrival to the router, if b, the number of cell buffers occupied is less than B_T, the incoming cell is queued for transmission. When $b > B_T$, all marked cells are discarded, while cells that are not marked are queued for

```
Procedure: Receive acknowledgment
        estimate_time := 2τ
        A_pos = fback := 0
        begin
           repeat
              if (ack = positive )                          Positive Acknowledgment
                 A_pos := A_pos + 1                  Increment positive acknowledgments
                 if( feedback = positive)
                    fback := fback + 1                  Increment positive feedback
                 if (W_u > 0)                                        Increment credits
                    W_u := W_u − 1
                 else if ( W_m > 0 )
                    W_m := W_m − 1
                 F_r := F_r + 1
                 if(message successfully transmitted)  Remove message
                    remove_message ()                    successfully transmitted
              else                                              Negative acknowledgment
                 if (W_m > 0)
                    W_m := W_m − 1
                 F_r := F_r + 1
              if (F_r ≥ W)                                    Adjust window parameters
                 if (adjust_flag = 0)
                    adjust_window_size()
                    adjust_flag := 1
                 else
                    adjust_flag := 0
                 fback := 0
                 F_r := 0
              if (current_time ≥ estimate_time)          Estimate excess bandwidth
                 W_e := estimate_excess()
                 estimate_time := current_time + 2τ
                 A_pos := 0
           until(doomsday)
        end
```

Figure 4.5: Receive acknowledgment

```
Procedure: estimate_excess()
        begin
              W_e := G x A_pos - e_fact x W
        end
Procedure: adjust_window_size()
        begin
              if ( fback > W/2)
                if (W < W_max )
                   W := W + 1
              else
                W := W/2
                if (W < W_min)
                   W := W_min
        end
```

Figure 4.6: Estimate excess and adjust window size routines

transmission. If the buffer is full, arriving cell that are not marked will replace marked cells in the buffer, otherwise, the arriving cell is discarded. The probability of cell loss is calculated from the number of marked cell discarded when $b > B_T$ and the number of unmarked cells that are discarded. The routine *compute_loss()* computes the probability of loss for a given buffer size B_k and buffer threshold B_T by accumulating the number of marked cells that exceed the threshold B_T and the number of cells that exceed the buffer size B_k in the array L_k. The probability of loss is then computed from the number of discarded cells L_k and the total number of cells transmitted. compute_loss() is shown in Figure 4.8. *cycle_start*, *regen_time*, *regen_count* and *cum_count* are used to implement regeneration. As discussed in Chapter 3, regeneration occurs at the end of a busy period. *cycle_start* is the beginning

of the first busy period while *regen_time* is the beginning of the second busy period. *regen_count* is the total number of cells received in the first busy period. *cum_count* is the total number of cells received in the first and the second busy periods. Using these variables, regeneration is performed by *regeneration()*, the routine shown in Figure 4.8.

The average queue length is calculated at the beginning of an arrival or transmission slot. We use the variable *timelast_change* to note the time when the average was computed last. The average is then computed by the function *avg()* , also shown in Figure 4.8.

Another performance measure used to analyze congestion at the router is the distribution of duration of congestion. The router is congested when b, the number of cell buffers occupied, is greater than B_T. We measure the distribution of congestion, as the time interval from the onset of congestion, that is when b exceeds B_T, to the time when there is no congestion. *start_cong* marks the onset of congestion. At the end of congestion, we compute the duration of congestion and increment the number of occurrences of the appropriate interval, buff_dist[i] by one. congestion_dist(), the routine which computes congestion histogram is given in Figure 4.8. Figure 4.9 is the cell transmission algorithm at a router. Transmission is fairly straightforward and may employ either a FCFS or a round robin service discipline for cells in the router buffer. Each user i is assigned a buffer $b[i]$. On arrival at the router, cells from user i are stored in $b[i]$. In the round robin service

discipline, one cell from $b[i]$, $0 < i \leq n$, where n is the number of active connections, is transmitted in a round robin fashion. In the FCFS case, the cell arrival time is noted, and the cells are transmitted according to their arrival time.

4.2 Network Architecture

First, we will give a description of the single node shown in Figure 4.10. This architecture will be used to test the LW mechanism. The choice of this architecture will be justified after its description. The router, R_0 shown in Figure 4.10 as a single server queue, has N users connected to it. Each user is connected to the router by an access line that is assumed to operate at the same speed as the trunk' line. The users are denoted by s_i, $0 < i \leq n$, the router is denoted by R_0 and the destination by D_0. We assume that an admission control mechanism that regulates the number of connections at any time to give the desired GOS is used in deciding whether to accept a new connection or not. For the purpose of simulation, a number of identical users, N, and an average cumulative rate of message arrival, λ, is dimensioned to give a desired network throughput, ρ. The router has a shared buffer of size B. The router and the destination are interconnected by unidirectional links with a one way propagation delay of τ. The propagation delay from the users to the router is zero. Each source generates traffic of the type described in

```
Procedure: Router receiver
    b := timelast_change := 0
    cycle_start := regen_time := 0
    cum_count := regen_count := 0
    start_cong := -1
    begin
       repeat
          if (b = 0 ) wait_event(cell_arrival)
          if( average_length > B_T)
              if (PTI = 000) PTI := 010
              else if (PTI = 001) PTI := 011
          if (b = 0 ) wait_event(cell_arrival)
          if ( b < B_T)                                    Queue all cells
            queue_cell()
            if (timelast_change < current_time)
               avg()
               timelast_change := current_time
            b := b + 1
          else if (b ≥ B_T and CLP = 1)                    Discard marked cells
            compute_loss()
            discard_cell
          else if (b < buffer_size )
            queue_cell()
            if(timelast_change < current_time)
               avg()
               timelast_change := current_time
            b := b + 1
          else if ( b = buffer_size )
            if ( marked cell in buffer)                    Check for marked cell in buffer
               discard_marked_cell
               queue_arriving_cell
               if (timelast_change < current_time)
                  avg()
                  timelast_change := current_time
               b := b + 1
            else                                           Discard unmarked cell
               discard_cell()
            compute_loss()
          else
            discard_cell()
            compute_loss()
          if (b ≥ B_T and start_cong = -1)                 Start of congestion
              start_cong := current_time
       until(doomsday)
    end
```

Figure 4.7: Router receiver

```
Procedure: compute_loss()
        begin
              L_k = L_k + 1
        end
Procedure: regeneration()
        begin
              cycle_start := regen_time
              regen_time := current_time
              cum_count := cum_count - regen_count
              regen_count := cum_count
        end
Procedure: congestion_dist()
        i = 0
        duration = 0                                   Congestion duration
        begin
             while (i < buf_dist_max) do
                  begin
                        if (current_time - start_cong > duration and
                            current_time - start_cong < duration + step )
                            buf_dist[i] := buf_dist[i] + 1
                            break
                        else
                            i = i + 1
                            duration = duration + step
                  end
        end
Procedure: avg()
        begin
              cum_count := cum_count + b
              average_length := cum_count / (current_time - cycle_start)
        end
```

Figure 4.8: Window update routines

```
Procedure: Router transmitter
begin
   repeat
      if( b = 0)
          wait_event(cell_arrival)                    Nothing to transmit
      transmit_cell()
      if (timelast_change < current_time)
          avg()
          timelast_change = current_time
      b := b − 1
      if (b < B_T) and (start_cong ≠ -1)              End of congestion
         congestion_dist()
         start_cong = -1
      if (b = 0 )
         regenerate()
   until(doomsday)
end
```

Figure 4.9: Router transmitter

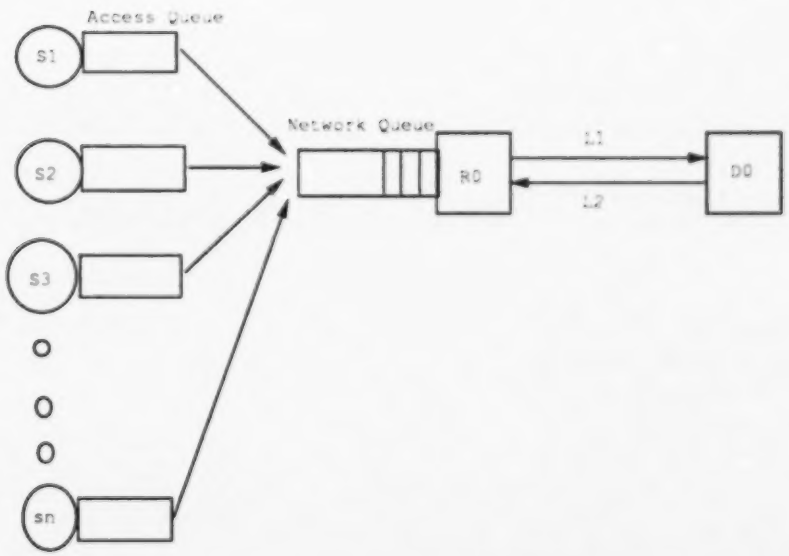

Figure 4.10: Congested Node

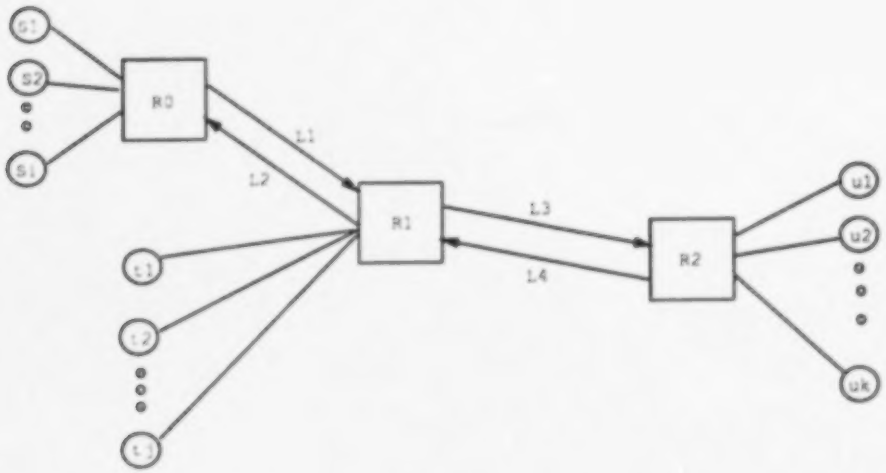

Figure 4.11: Three node network

Section 4.1.2. Traffic generated by a source is queued at the network node at an input speed equal to the access line speed as long as the number of frames transmitted by the user that have not been acknowledged does not exceed the W. When the number of unacknowledged frames equals W, arriving frames are queued at the access queue until additional credits become available. The access queue is assumed to be of an infinite size. Our objective is to measure the performance of router R_0, which may experience congestion because of traffic from the users.

To justify the use of a single node as a good approximation for testing the LW mechanism, let us consider the network with a configuration shown in Figure 4.11. This network has three routers, R_0, R_1 and R_2, represented by rectangles. Router R_0 is connected to users $s_0, s_1, ..., s_i$, and R_1 is connected

to users $t_0, t_1, ..., t_j$, and to router R_0 by links L_1 and L_2. Router R_2 is connected to users $u_0, u_1, ..., u_k$, and to router R_1 by links L_3 and L_4. Each of the routers R_0, R_1 and R_2 may become congested because of traffic from the users and links from other routers connected to it. Let us consider what happens at router R_1 during congestion. Suppose congestion is because of traffic whose virtual circuits are routed through L_3. Traffic from L_4 whose destination is either s_k or t_k will not affect congestion at R_1 since all arriving cells from L_4 need not be buffered at R_1 if the links on which these cell are to be transmitted are not congested. Traffic from link $L1$ that are to be routed through L_3 will contribute to congestion at R_1. L_1 may however be considered as one of the sources directly connected to R_1. Call it t_{j+1}, with an access link operating at the same line speed as L_1 and carrying traffic whose characteristics match those of the superposition of the process of sources s_1 through s_i. Router R_2 operates at the line speed of L_3. Transfer of cells from L_3 to the router R_2 therefore, does not introduce additional congestion at R_1. The single congested node of Figure 4.10 can therefore approximate a congested node such as R_1 with the appropriate traffic process chosen for each source.

4.2.1 Dynamic window mechanism

The dynamic window mechanism is implemented by turning off the estimation and transmission of excess cells.

4.2.2 VLB

The implementation of the VLB is similar to the LW except for the flow control technique, i.e. the assignment of credits and excess capacity that a user is allowed to utilize. In the VLB, when the number of credits is exhausted, transmission is not suspended. In this case, all arriving frames are marked and transmitted. The main difference between the LW and the VLB is therefore in the method of credit assignment and the excess capacity that may be utilized by a user. In the LW, the number of excess frames is limited to W_e, while in the VLB, there is no limit.

Credit assignment in the VLB is rate-based. The definition of "rate" may be one of the following: each credit may permit transmission of one frame or one cell in an average interval of time. The average interval over which a credit is assigned is based on the negotiated transmission rate, and this interval will be referred to as the *credit-interval*. Therefore, the "rate" specifies the number of cells or frames that may be transmitted within a credit-interval. When the "rate" specifies the number of cells that may be transmitted in a credit-interval, it will be referred to as cell-rate, otherwise,

Rate	Frame-rate	Cell-rate	Frame-rate	Cell-rate
Buffer threshold B_T	200	200	300	300
Utilization	0.7200	0.8834	0.7139	0.8160
Throughput	0.7023	0.4293	0.7023	0.5215
%Retransmission	6.0%	86.2%	3.92%	79.9%
Marked frames received	52.0%	47.7%	52.7%	52.2%
Mean end-to-end delay (Cells)	2730	15681	5281	9225
Maximum Buffers	417	363	471	516

Table 4.1: Rate based on cells and frames per credit-interval (VLB)

if it specifies the number of frames, it will be referred to as frame-rate.

The need to distinguish between cell-rate and frame-rate arises because a user subdivides a frame into ATM cells before transmission. Control of traffic injection into the network is implemented by the user by transmitting cells whose number is equal to that of the available credits without marking. All cells transmitted when credits are exhausted are marked. Thus the definition of "rate" influences transmission of marked and unmarked cells within a frame. Using either frame-rate or cell-rate, strategies A and B of frame transmission already discussed for the LW in Chapter 3 may arise.

To decide on which definition of rate to use, we experimented with frame-rate as well as cell-rate using strategy A. The results are presented in Table 4.1. The experiments were performed at a load of 0.7 using a round robin service discipline at the router.

Inspecting the percentage of retransmissions, the throughput and the

average end-to-end frame delay for $B_T = 200$ and $B_T = 300$ reveals that frame-rate is superior to cell-rate. This performance may be due to the following reasons. When some of the cells in a frame are marked while others are not marked, there is no guarantee that a complete frame will arrive at its destination. The VLB does not limit the number of marked cells transmitted. A high rate of discarding of marked cells means that the frames with cells that were unmarked may also have to be retransmitted, and the credits used in transmitting the unmarked cells are wasted, hence, the long end-to-end delay. The high percentage of discarding marked cells at the router and the subsequent retransmission of incomplete frames that are received at the destination, is also reflected in the utilization and the throughput of the transmission link. Note that since the frame size used in this simulation is not fixed, assigning average rates which actually hold after a large number of transmissions is not fair, since this average may not hold on a frame by frame basis. We shall therefore use frame-rate for the comparison of the performance of the VLB to that of the LW mechanism.

4.2.3 Network specification

In this section we define the parameters used in the simulation model. These are the traffic generation parameters, window control parameters and the network characteristics. Links L_1 and L_2 in the single node model shown in

Figure 4.10 are unidirectional, and operate at 45 Mbits/s, with a propagation delay τ = 20 milliseconds. This is approximately 2200 cells (one cell = 424 bits). We use β = 2000 cells, α = 17857 cells and γ = 111 cells, to generate traffic with a burstiness factor equal to 10. The number of users N is chosen to achieve the desired throughput ρ. We test the LW under throughput values of $\rho = 0.7$ and $\rho = 0.8$. This throughput is carried out for N = 70 and N = 80 users, respectively.

The results to be presented are obtained using the following control parameters. A window reduction factor, r_{fact} equal to 0.5 that has been chosen based on results in [2] which compare the probability of loss for different reduction factors. The initial window size, W_{in} = 3 frames, the minimum window size, W_{min} is equal to 1 frame, the estimation factor $\epsilon_{fact} = 0.0$, and the gain G = 2. These parameters were chosen after a number of trials to determine values that give a reasonable performance. Simulation time limits the number of factors that may be tested, so the results presented are not exhaustive.

First the performance of the window mechanism is compared to the LW under the same condition. It would also be interesting to know how the LW performs compared to rate-based control mechanisms such as the VLB. The VLB is also chosen because it is one of the most promising congestion control techniques in broadband networks [22]. Therefore, we compare the performance of each of the LW network configurations to the VLB. The

VLB is simulated with credits assigned at an average rate to each user, the values being 1 frame in 1100 units (cells) per user. Each user is permitted to accumulate a maximum of $Q = 10$ credits, which accommodates traffic with a burstiness factor of 10.

4.3 Results

We measure the probability of cell loss at the router, mean end-to-end frame delay, and the link throughput using the LW congestion control mechanism. We also measure the distribution of end-to-end frame delay and the distribution of buffer occupancy during congestion at the router. Each of the performance measures of the LW is compared to that of the VLB. Results for the FCFS as well as the round robin (RR) service disciplines at a load of 0.8 are presented. Only the RR service discipline is examined under load levels of 0.7, because, as will be seen in the results obtained at this load using the RR service discipline, congestion is not significant and it is unlikely that we would observe anything different using the FCFS service discipline.

Results for the two frame transmission strategies, namely, allowing transmission of a mixture of marked and unmarked cells within a frame, which is referred to as strategy A and the case where such a mixture is not allowed, referred to as strategy B, are presented. In the tables, we denote the results of strategy A by FCFS-A and RR-A, and for the B strategy, we use FCFS-

B and RR-B. For the graphs, the labels LW-A and VLB-A denote results obtained using strategy A while the labels LW-B and VLB-B denote results obtained using strategy B. To obtain probability of loss in the order of 10^{-6}, simulation runs with more than 10,000,000 transmitted cells were performed.

First we show the performance of the window mechanism in Figure 4.12. We observed that the probability of loss using the window mechanism is much higher than the LW. Table 4.2 is a summary of other performance measures. These results indicate two things:

1. The lack of sufficient credits and hence low throughput and high end-to-end delay of the window mechanism (Table 4.2).

2. The slow response of the window mechanism resulting in a high probability of loss (Figure 4.12).

For the remainder of this chapter, we shall be considering the LW mechanism only.

Results for a 0.7 load level

Tables 4.3 and 4.4 summarize the observations made for the LW and the VLB at a load level of 0.7, respectively. Results for $B_T = 100$ and $B_T = 200$ are the only ones shown at this load level, since, at higher values of B_T, congestion is not detected. First, comparison of results of RR-A and RR-B in Table 4.3 indicates that for $B_T = 100$, RR-B has a higher throughput. The higher

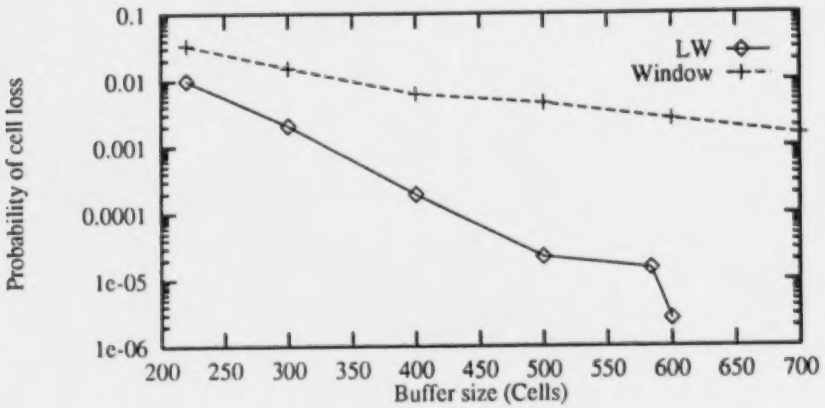

Figure 4.12: Probabilty of loss for LW and Window mechanisms , for B_T= 200, Load =0.8

Mechanism	Window	LW
Service discipline	RR	RR
Buffer threshold	200	200
Initial window size	3	3
Minimum window size	1	1
Estimation factor	0.0	0.0
Reduction factor	0.5	0.5
Utilization	0.6908	0.7627
Throughput	0.6893	0.7421
Mean end-to-end delay (cells)	7212	4319
Buffer size	600	580

Table 4.2: Summary of results for the Window and LW mechanisms for ρ = 0.8

throughput in RR-B may be because of loss of credits in RR-A, which may happen as follows. When a mixture of marked and unmarked cells within a frame are transmitted, the entire frame may have to be retransmitted should any of the marked cells be discarded at the router. In addition to having to retransmit a frame that is received at the destination with some cells missing, the credit used to transmit unmarked cells is wasted since the entire frame including the unmarked cells will have to be retransmitted. The fact that the lower throughput in RR-A occurs only at $B_T = 100$ is not surprising since, at this threshold, there is a higher chance of a marked cell being discarded.

For $B_T = 100$, the percentage of retransmission in RR-B is higher in RR-A. The higher throughput in RR-B compared to RR-A despite a higher percentage of retransmission is because of the following reason. Link utilization in both transmission strategies is less that the transmission link capacity. In RR-B it is equal to 0.7194, which means 0.2806 of the transmission capacity is unused. Retransmission may utilize the unused capacity without causing a decrease in throughput as might be expected. At this same value of threshold, the mean end-to-end delay in RR-B is higher than in RR-A. This confirms the fact that in RR-B, more retransmitted frames are received at the destination than in RR-A.

At $B_T = 200$, a higher throughput is attained by RR-A compared to RR-B, but with a higher percentage of retransmission. The percentage of retransmitted frames at $B_T = 200$ is about 50% less than that retransmitted

at $B_T = 100$.

For RR-A, the throughput and the percentage of frames received at the destination that are marked are higher at $B_T = 200$ than at $B_T = 100$. The percentage of retransmissions is lower for the respective value of threshold. A higher percentage of marked frame successfully transmitted has a two fold effect. First, it increases throughput and secondly, it decreases the percentage of retransmission. As expected in RR-B, the percentage of retransmissions at $B_T = 200$ is lower than at $B_T = 100$, while the percentage of frames received at the destination that are marked is higher. The average end-to-end delay increases with the number of retransmissions, which is also expected.

Results of the VLB in Table 4.4 show no significant difference between the performance of RR-A and RR-B. All users are allocated the required transmission capacity when required, and because of little or no congestion, the frame transmission strategy does not affect performance.

In these two tables, the percentage of retransmitted frames by the LW is of the same order of magnitude as that of the VLB. The VLB however, attains a higher throughput at $B_T = 100$ in RR-A than the LW. This may be because of the fact that there is no limit to the number of excess cells that may be transmitted by the VLB unlike the LW which limits the number of marked cells. Therefore, despite a similar percentage of cells that are discarded, the VLB does not suffer delays similar to that of the LW which has to wait for credits in the form of arrivals of acknowledgments to frames

Service discipline	RR-A	RR-B	RR-A	RR-B
Buffer threshold	100	100	200	200
Initial window size	3	3	3	3
Minimum window size	1	1	1	1
Estimation factor	0.0	0.0	0.0	0.0
Reduction factor	0.5	0.5	0.5	0.5
T_{reset}	10	10	10	10
Utilization	0.6581	0.7194	0.7124	0.7043
Throughput	0.6348	0.6925	0.6977	0.6923
%Retransmission	10.3%	12.1%	6.5%	5.9%
%Marked frames received	41.0%	38.9%	43.9%	44.1%
Mean end-to-end delay (cells)	4600	5004	3863	3755
Buffer size	500	500	500	500

Table 4.3: Summary of result for LW for $\rho = 0.7$

transmitted, some of which may have been discarded. The average end-to-end delay at both the threshold values for the VLB is lower than for the LW. The percentage of marked frames received at the destination using the VLB is higher than that for the LW mechanism; an indication that the VLB relies more heavily on the successful transmission of marked cells compared to the LW. For the same values of B_T and buffer size at the router, the results presented in Tables 4.3 and 4.4 show that the maximum number of buffers used by the LW is equal to the buffer size while for the VLB, it is less than the buffer size. This, again may be attributed the availability of credits in the LW mechanism. In the VLB, there is no backlog of traffic waiting for transmission because of there is no shortage of credits, which may happen

Service discipline	RR-A	RR-B	RR-A	RR-B
Buffer threshold	100	100	200	200
Utilization	0.7274	0.7292	0.7106	0.7144
Throughput	0.6933	0.6966	0.6927	0.6994
%Retransmission	11.2%	10.9%	7.2%	5.2%
% Marked frames received	50.3%	50.2%	51.5%	52.1%
Mean end-to-end delay (cells)	3117	3111	2994	2708
Maximum buffers	324	394	378	400

Table 4.4: Summary of result for VLB for $\rho = 0.7$

in the LW scheme. If credits become available to users that are backlogged at the same time, simultaneous transmission may results in a burst of traffic to arrive at the router which may cause overflow of the buffer size in the LW. This is less likely to happen in the VLB. Thus the maximum number of buffers that may be occupied in the LW is higher than in the VLB.

Figures 4.13 and 4.14 show the probability of loss for increasing buffer sizes at a fixed buffer threshold, B_T, equal to 100 and 200, respectively. In these two figures, the VLB exhibits a lower probability of loss. Inspection of the graphs labelled LW-A and LW-B in the two figures shows that graphs labelled LW-A have a lower probability of loss. Graphs labelled VLB-A and VLB-B do not show a consistent behavior, as is seen in the two figures.

To understand the operational aspects of both the LW and the VLB, we shall examine the distribution of duration of congestion and the distribution of end-to-end delay. Figure 4.15 shows the distribution of buffer occupancy

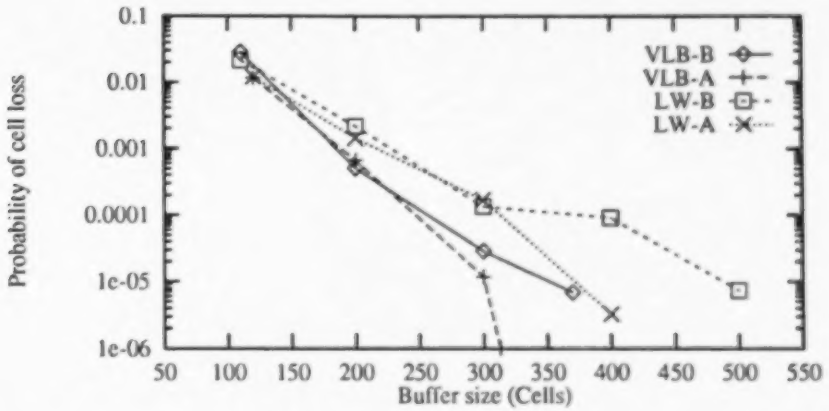

Figure 4.13: Comparison of probability of loss for LW and VLB for $B_T = 100$, Load = 0.7

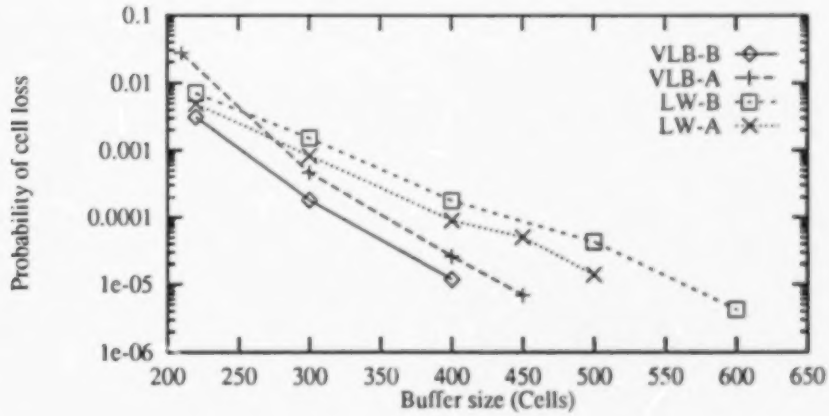

Figure 4.14: Comparison of probability of loss for LW and VLB for $B_T = 200$, Load = 0.7

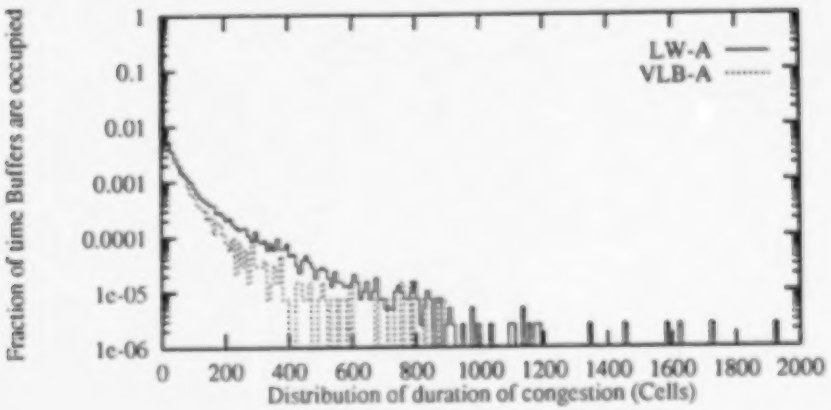

Figure 4.15: Distribution of duration of congestion for LW-A and VLB-A for B_T= 100, Load =0.7

at $B_T = 100$. In this figure, it is clear that the duration of the congestion for the VLB-A is shorter than for the LW-A.

Figure 4.16 shows the distribution of the end-to-end delay at $B_T = 100$. In this figure, both the LW-A and the VLB-A have the same range of delay and in addition, the VLB-A exhibits a notable characteristic that we shall now point out as it recurs in the distribution of end-to-end delay at all other levels of load and buffer threshold values.

We observe in Figure 4.16 that the arrival of frames at the destination when the VLB-A is used can be divided into two groups. The first group of frames arrive at the destination after a delay of τ (equal to 2200). Those are the frames with a delay in the range 0-3000 cells in Figure 4.16. The rest

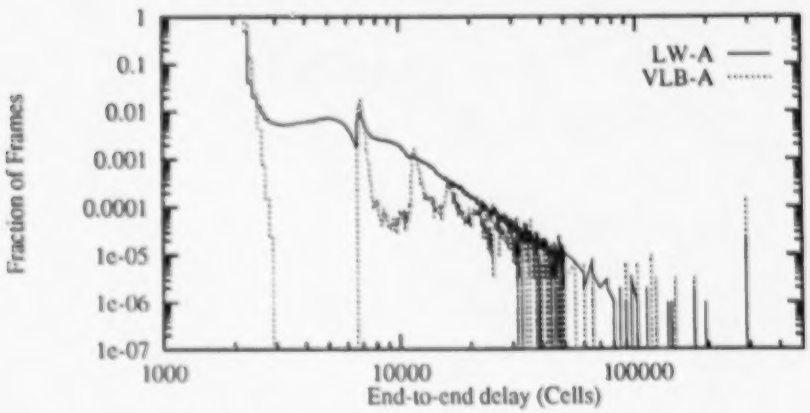

Figure 4.16: Distribution of end-to-end delay for LW-A and VLB-A for B_T= 100. Load =0.7

of the frames arrive at the destination after a delay of 3τ (> 6000), which is the minimum delay experienced by a frame that is retransmitted. This behavior should be expected from the VLB-A because it permits all frames to be transmitted upon generation, with some slight delay because of access line speed. Some of the frames are transmitted using the available credits, and others are marked and transmitted. Some of the marked cells may be discarded, and the frames that such cells belong to will be retransmitted when a negative acknowledgment is eventually received from the destination. The shortest time a retransmission request may be received is 2τ, the round trip propagation delay. Therefore, a retransmitted frame may arrive at the destination after 3τ, from the time of generation. This is why in Figures 4.16,

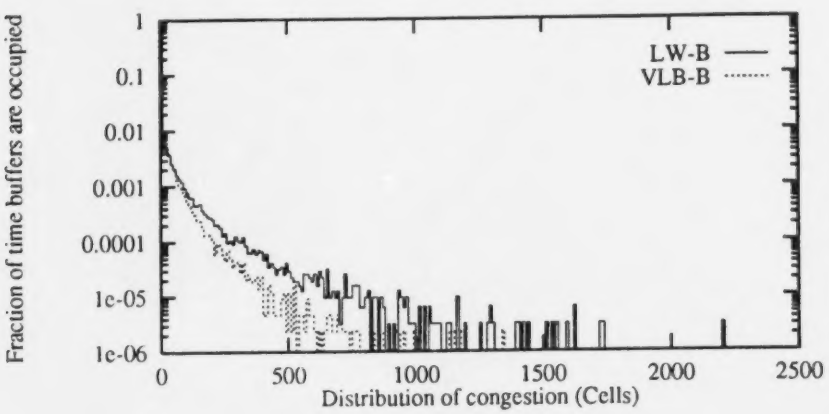

Figure 4.17: Distribution of duration of congestion for LW-B and VLB-B for B_T= 100, Load = 0.7

there is a gap in the range of delay that frames experience in the VLB-A. The LW mechanism does not explicitly exhibit a similar behavior, an indication that there isn't a sufficient number of credits to transmit all frames upon generation.

Figures 4.17 and 4.18 are for frame transmission strategy B. These figures have a similar pattern as Figures 4.15 and 4.16, except in Figures 4.17, the maximum duration congestion of LW is higher.

Figures 4.19 and 4.20 are the distribution of duration of congestion and the distribution of end-to-end delay at B_T = 200, for strategy A. These results are similar to those obtained for B_T = 100. Figures 4.19 shows that the duration of congestion for the VLB-A is shorter than for the LW. The

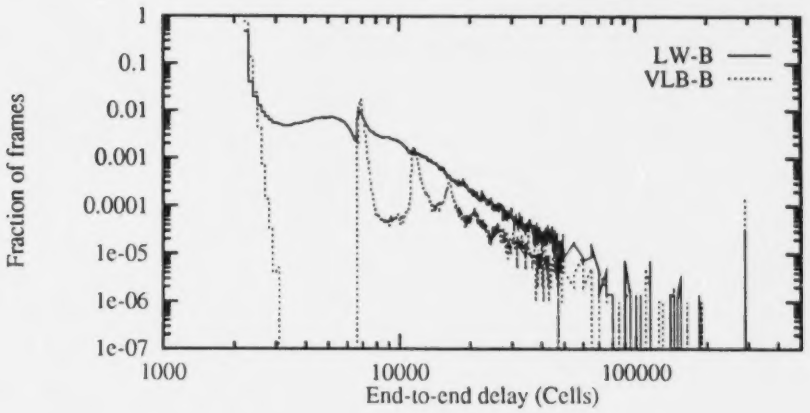

Figure 4.18: Distribution of end-to-end delay for LW-B and VLB-B for B_T= 100, Load = 0.7

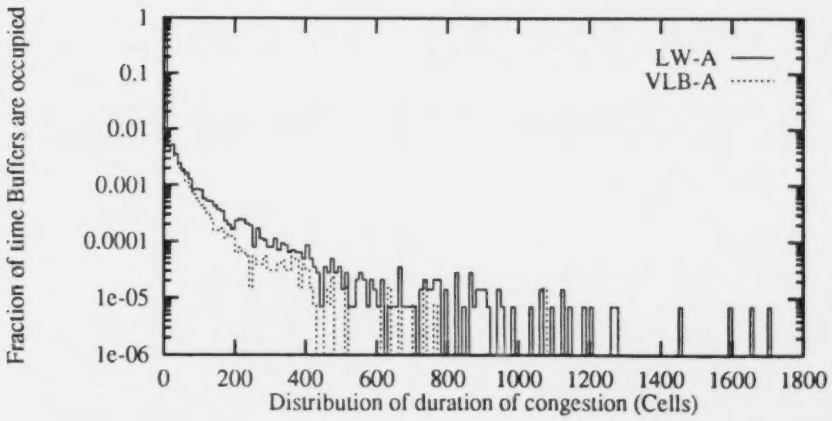

Figure 4.19: Distribution of duration of congestion for LW-A and VLB-A for B_T= 200, Load =0.7

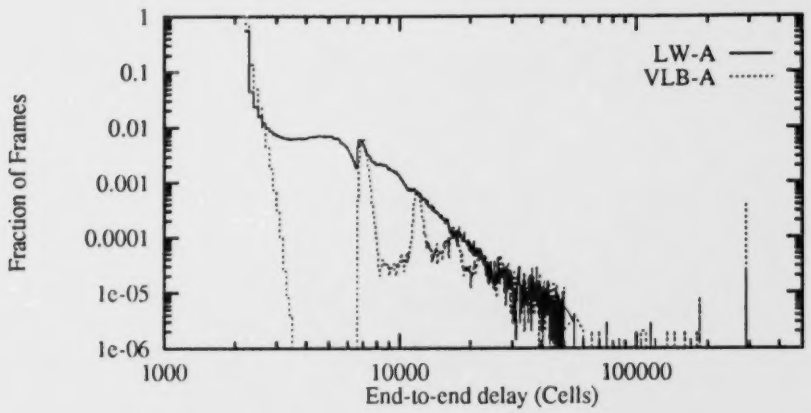

Figure 4.20: Distribution of end-to-end delay for LW-A and VLB-A for B_T= 200, Load =0.7

distribution of the end-to-end delay in Figure 4.20 shows that frames may take longer to reach the destination than the measured interval for both the VLB-A and the LW-A.

Figures 4.21 and 4.22 are for observations made using transmission strategy B. These figures are similar to Figures 4.17 and 4.18

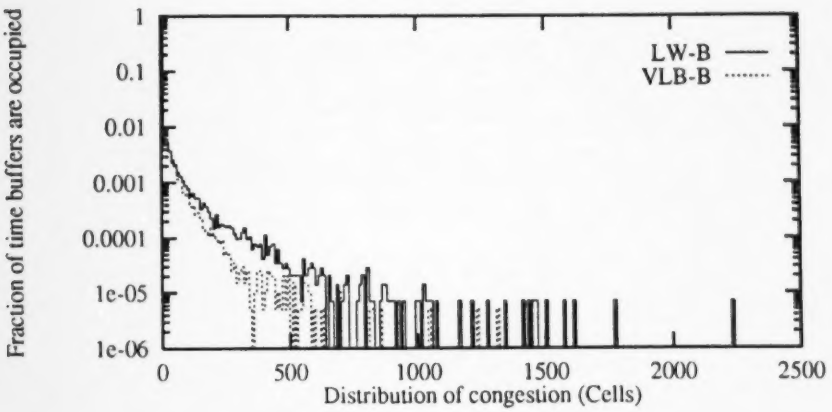

Figure 4.21: Distribution of duration of congestion for LW-B and VLB-B for B_T= 200, Load = 0.7

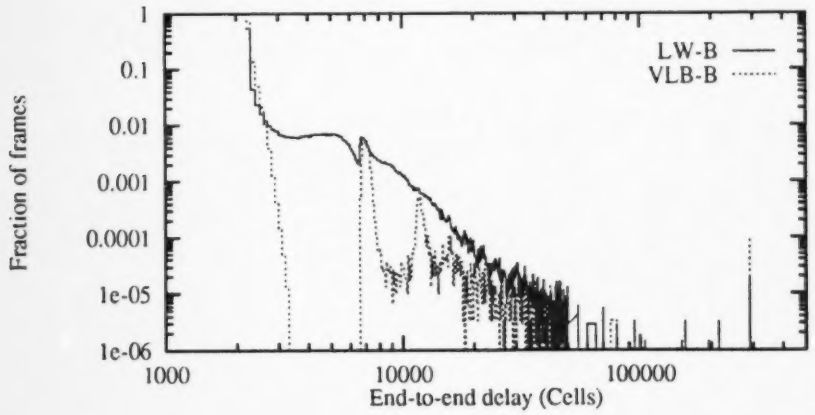

Figure 4.22: Distribution of end-to-end delay for LW-B and VLB-B for B_T= 200, Load = 0.7

Results for $\rho = 0.8$

When the load level reaches 0.8, the degree of congestion is higher than at a load level of 0.7. The results obtained at this load level, are therefore more indicative of the congestion control ability of the mechanism. Results for the two frame transmission strategies are presented under the titles RR-A and FCFS-A for strategy A and RR-B and FCFS-B for strategy B.

Table 4.5 is a summary of observations for the LW mechanism using the FCFS-A and RR-A service discipline at values of B_T equal to 200 and 300. At B_T = 200, FCFS-A has a lower throughput than FCFS-B. At this same value of threshold, RR-A also has a lower throughput than RR-B. The lower throughput in FCFS-A and RR-A compared to FCFS-B and RR-B, respectively, may be because of similar reasons given for RR-A having a lower throughput than RR-B at the load level of 0.7. In FCFS-A and FCFS-B, the throughput, percentage of retransmitted frames and the mean end-to-end delay are approximately equal to those in RR-A and RR-B, respectively. This indicates that the LW is not sensitive to the service discipline at the router.

At a higher value of threshold, B_T = 300, although performance in terms of throughput and retransmissions in FCFS-B are better than those in FCFS-A, the performance of FCFS-A is closer to FCFS-B at this threshold compared to result obtained at B_T = 200. The performance in RR-B at this

Service discipline	FCFS-A	FCFS-B	RR-A	RR-B	FCFS-A	FCFS-B	RR-A	RR-B
Buffer threshold	200	200	200	200	300	300	300	300
Initial window size	3	3	3	3	3	3	3	3
Min window size	1	1	1	1	1	1	1	1
Estimation factor	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Reduction factor	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
T_{reset}	10	10	10	10	10	10	10	10
Utilization	0.7689	0.8151	0.7627	0.8198	0.8082	0.8101	0.7814	0.8175
Throughput	0.7474	0.7905	0.7421	0.7956	0.7885	0.7933	0.7659	0.8003
%Retransmission	9.4%	11.8%	8.9%	11.7%	8.7%	6.8%	6.2%	8.3%
%Marked frames	41.7%	39.4%	41.9%	39.6%	42.2%	43.6%	44.0%	42.1%
Mean end-to-end delay (cells)	4336	4945	4468	4898	4120	4283	3988	4241
Buffer size	580	600	580	600	670	700	700	700

Table 4.5: Summary of results for LW for $\rho = 0.8$

threshold is also better than that of RR-A. The lower throughput in RR-A and FCFS-A compared to RR-B and FCFS-B, respectively, may again be because of cell discarding which results in wastage of credits when a mixture of marked and unmarked cell are transmitted within a frame. Some delay may be incurred before the destination detects the loss of cells in a frame which results in additional delay before a NAK arrives at the source.

For threshold of 300, RR-B has a throughput that is higher than RR-A yet the percentage of retransmissions in RR-B is higher than in RR-A. This can be explained in a similar way as at the load value of 0.7. As in the case of the load level of 0.7, the utilization is lower than the transmission capacity of the links. Retransmission may therefore utilize the unused capacity without causing a degradation in throughput.

Table 4.6 is a summary of observations made for VLB at a load level of

0.8. also using the FCFS and RR service discipline at buffer threshold shown in the table. First, the FCFS and RR service discipline show a notable difference, especially the percentage of retransmissions and the mean end-to-end delay. From this observation, we can say that the VLB is more sensitive to the service discipline at the router than the LW.

At $B_T = 200$, the percentage retransmissions, and the mean end-to-end delay of the FCFS-A is better than that of RR-A. The higher percentage of retransmission in RR-A as compared to FCFS is because of having more cells discarded in RR-A. FCFS-A transmits cells in the order of their arrival at the router. Using FCFS-A. marked cells queued for transmission will have a higher chance of being transmitted than RR-A for the following reason.

In RR-A. cells already queued for transmission at the router may be superseded by arrivals in channels that have shorter queues at the router. Therefore, marked cells queued at the router are more liable to being discarded than in the FCFS-A. For example, if the buffer becomes full, a marked cell that has been waiting for service long enough may be discarded in order to accommodate an arriving unmarked cells in the RR-A. Thus in the RR-A, the longer a marked cell has to wait for service at the router, the higher the probability of being discarded.

At $B_T = 200$, results for FCFS-A are worse than those for FCFS-B. The result for RR-A are also worse than those for RR-B, for the same value of threshold. This may be because of wastage of credits by strategy A, when a

Service discipline	FCFS-A	FCFS-B	RR-A	RR-B	FCFS-A	FCFS-B	RR-A	RR-B
Buffer threshold	200	200	200	200	300	300	300	300
Utilization	0.9384	0.9257	0.9685	0.9037	0.9711	0.8595	0.8641	0.8762
Throughput	0.7899	0.8030	0.7840	0.7990	0.7917	0.8000	0.7934	0.8003
%Retransmission	55.1%	50.6%	69.5%	45.4%	64.7%	26.0%	32.0%	35%
% Marked frames	38.0%	39.5%	30.0%	41.7%	32.7%	47.5%	46.6%	45%
Mean end-to-end delay (cells)	14573	13006	23197	10532	20694	5517	6832	7958
Buffer size	600	600	600	600	700	700	700	700

Table 4.6: Summary of results for VLB for $\rho = 0.8$

mixture of marked and unmarked cells within a frame are transmitted. This is similar to the observation made at a load level of 0.7. At $B_T = 300$, except for FCFS-A, there is an improvement in performance of all schemes compared to their corresponding values at $B_T = 200$. The results of the FCFS-A at this threshold is unexpected. This may be due to the method of cell choice for discarding when an unmarked cell arrive at the router when the buffer is full. A simple discarding procedure that removes the first marked cell at the head of the queue is used. This can have an adverse effect on the FCFS service discipline. Further experimentation is required in order to fully explain this.

Figures 4.23 and 4.24 show the probability of cell loss for increasing buffer size at $B_T = 200$ and $B_T = 300$, respectively, for both the LW and VLB-A. In both of these figures, we observe that the LW-A has a lower probability of loss than the VLB-A. We also notice in both the figures that as the buffer sizes get larger, the probability of loss labelled LW-A and VLB-A approach a

bound that is lower that than LW-B and VLB-B respectively. It appears that by using frame transmission strategy A, the buffer size required for a bounded loss is smaller than that of frame transmission strategy B. The observation is made at a buffer size 600 for $B_T = 200$ and at a buffer size of 700 for $B_T = 300$. Once the number of cells in the router buffer exceeds the buffer threshold, the cells admitted into the buffer are all unmarked. For the buffer to overflow, a number of users must transmit unmarked cells within an interval less than $\tau/2$ ($\tau = 2200$). This suggests some kind of "synchronized" transmission of unmarked cells by a number of users with cumulative transmission capacity in excess the buffer size at the router. The chance for such an occurrence seems to be higher for strategy B than strategy A. This may be because strategy A permits a mixture of marked and unmarked cells within a frame, which reduces the chances of unmarked cells occupying buffers in excess of the size mentioned above.

The distributions of buffer occupancy during congestion are shown in Figures 4.25 and 4.26 for $B_T = 200$ and $B_T = 300$, respectively, for the transmission strategy A. In this two figures, the maximum duration of congestion of LW-A is smaller than that of VLB-A. Figures 4.27 and 4.28 which are the distribution duration of congestion for frame transmission by strategy B also show that the maximum duration of congestion of LW-B is marginally smaller than that of VLB-B. Comparing the duration of congestion for the LW in Figures 4.25 and 4.26 and Figures 4.27 and 4.28 it is clear that by

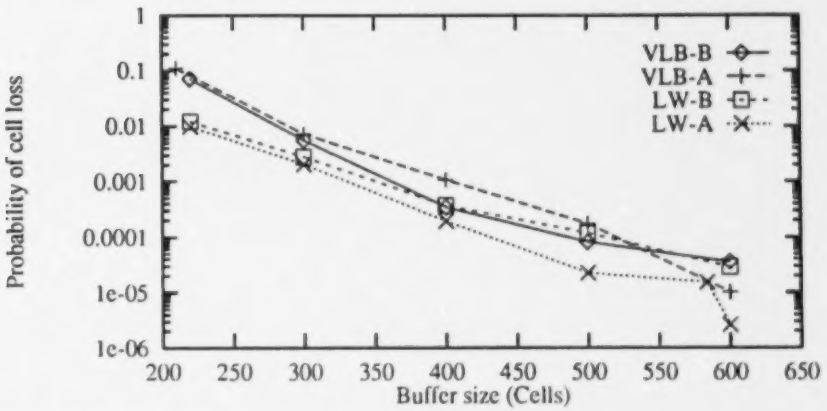

Figure 4.23: Comparison of probablity of loss for LW and VLB for B_T= 200, Load = 0.8

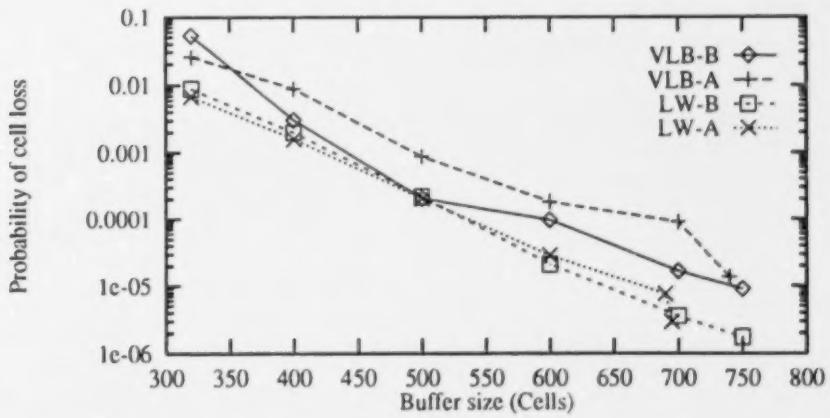

Figure 4.24: Comparison of probablity of loss for LW and VLB for B_T= 300, Load = 0.8

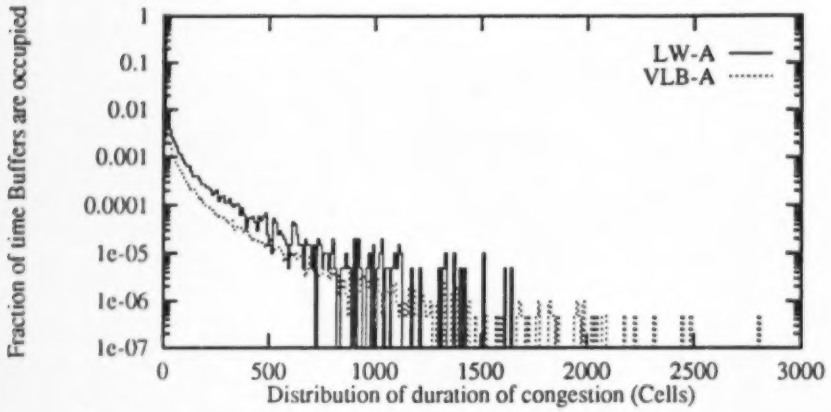

Figure 4.25: Distribution of duration of congestion for LW-A and VLB-A for B_T= 200, Load = 0.8

using strategy A, the maximum duration of congestion is smaller than using strategy B. The VLB does not have a similar trend.

And finally, the distribution of end-to-end delay for strategy A are shown in Figures 4.29 and 4.30. Figures 4.31 and 4.32 the distribution of end-to-end delay for strategy B. This figures follow a similar pattern as those of load level of 0.7 discussed earlier. We note here that the end-to-end delay of some frames exceed the measurement interval of about (100τ). In all these figure, we observe that the percentage of frames that exceed the measurement interval is lower for the LW.

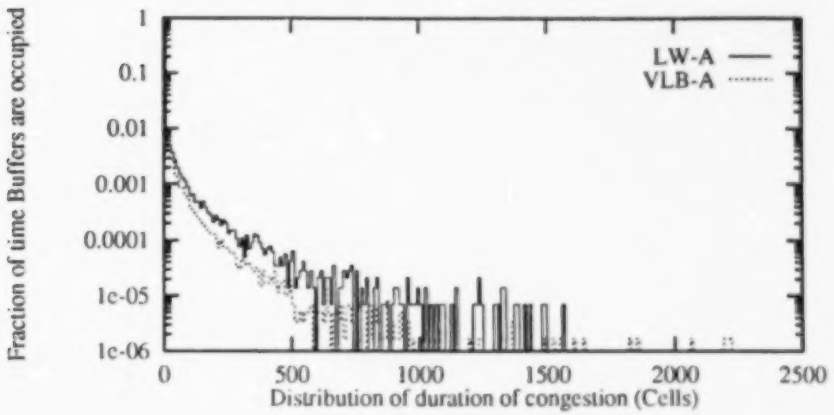

Figure 4.26: Distribution of duration of congestion for LW-A and VLB-A for $B_I = 300$, Load = 0.8

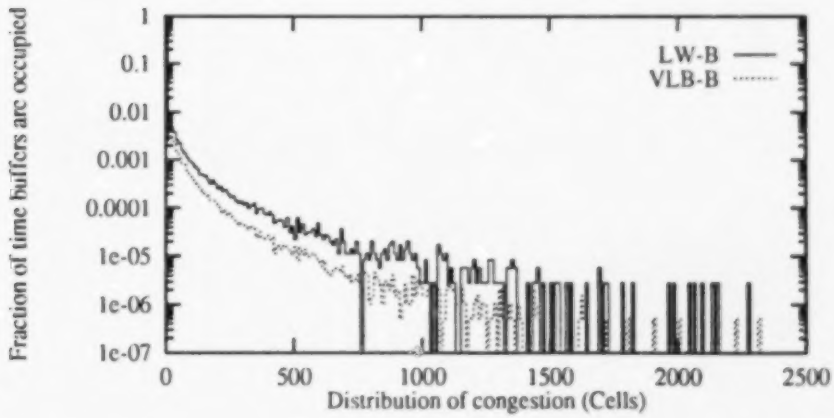

Figure 4.27: Distribution of duration of congestion for LW-B and VLB-B for $B_I = 200$, Load = 0.8

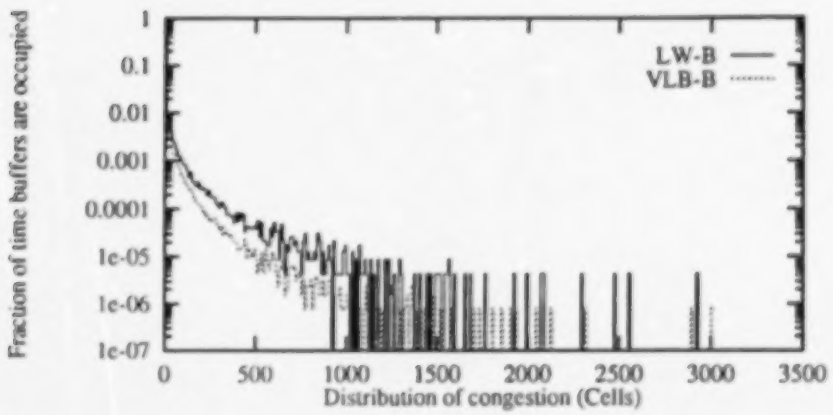

Figure 4.28: Distribution of duration of congestion for LW-B and VLB-B for $B_T = 300$, Load = 0.8

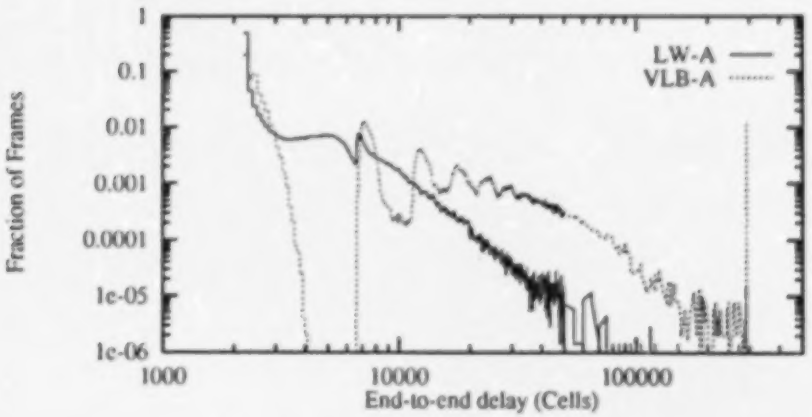

Figure 4.29: Distribution of end-to-end delay for LW-A and VLB-A for $B_T = 200$, Load = 0.8

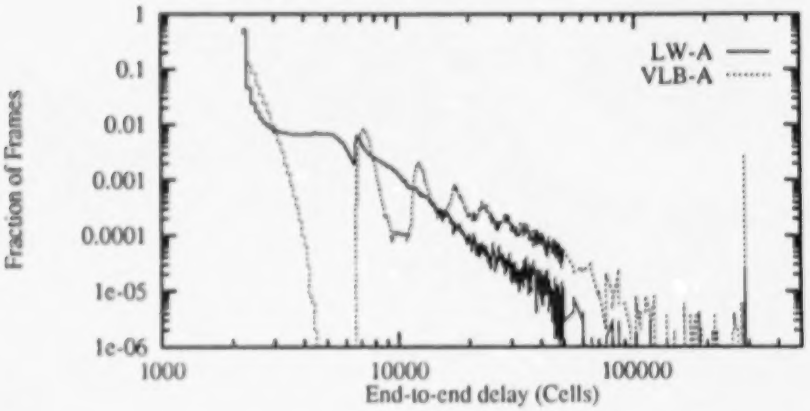

Figure 4.30: Distribution of end-to-end delay for LW-A and VLB-A for B_T= 300, Load = 0.8

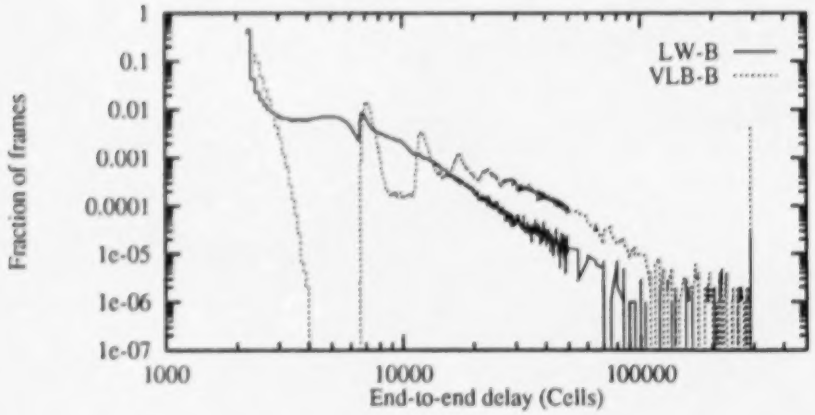

Figure 4.31: Distribution of end-to-end delay for LW-B and VLB-B for B_T= 200, Load = 0.8

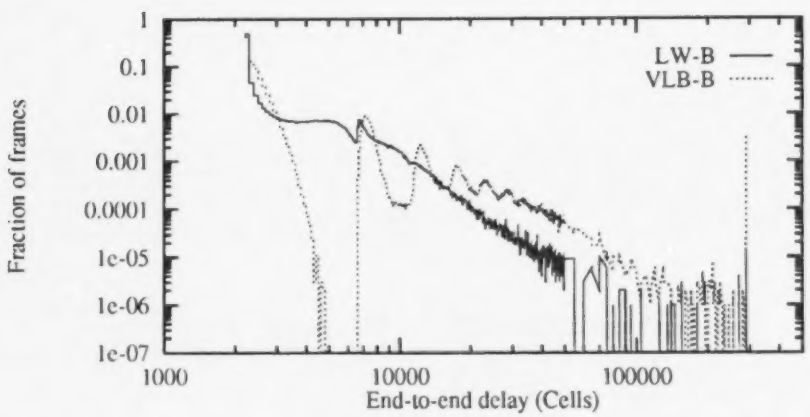

Figure 4.32: Distribution of end-to-end delay for LW-B and VLB-B for B_T= 300, Load = 0.8

4.3.1 Response to change

In this section the results of responsiveness to load changes of the LW are presented. Only strategy A of frame transmission is used in this test. The response to load changes of both the LW and the VLB are tested by simulating a network in which the offered load varies with time. The result presented here is for the case where the offered load is decreased from a value of 0.8 to 0.4. The network is allowed to run at the reduced load until a steady value of throughput is attained, then the load is set to the original value of 0.8.

Results for this test are given in Figures 4.33, 4.34 and 4.35. The throughput curve shown in Figures 4.33 show that the LW mechanism responds to changes at the same rate as the VLB but with some time lag when the load increases. This time lag may be because of the method used to obtain the results. The changes in offered load is made after a fixed number of frames is successfully transmitted. The graph in Figures 4.33 is a plot of time versus throughput. The actual time when the offered load changes for the two mechanisms may not be the same, therefore we cannot draw a definite conclusion from the relative transitions of offered load from high to low and back to high again, for the two mechanism. Figures 4.34 shows that the average queue length at the router at the reduced load value is lower for the VLB than the LW. Both mechanisms reach their minimum queue length value at about the same time indicating a faster rate of change for the LW compared to

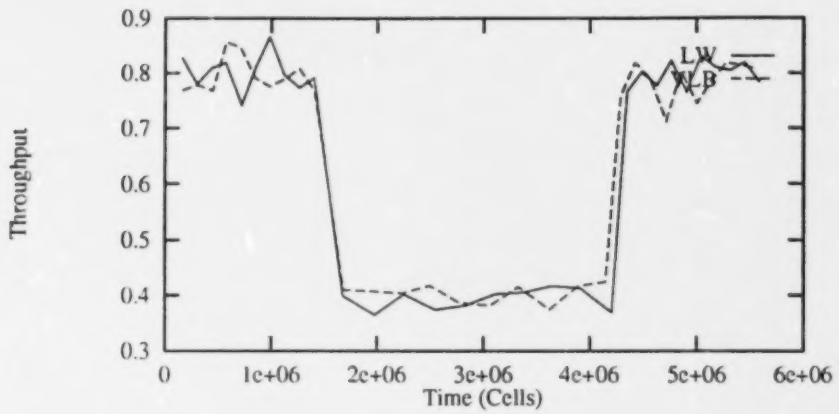

Figure 4.33: Througput variation with time for LW and VLB for B_T= 200, Load =0.8

the VLB. The response when the load increases are almost the same for the two mechanism, but the LW settle at a lower average queue length than the VLB. The end-to-end delay in Figure 4.35 show that at low load, the VLB deliver frames with a minimum delay. Delay of the LW may be attributed to insufficient credits.

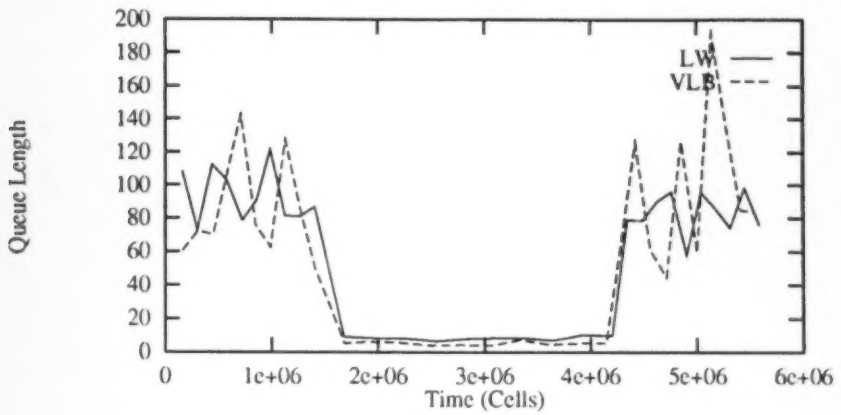

Figure 4.34: Mean queue length variation at router over time for LW and VLB: $B_T = 200$. $\rho = 0.8$

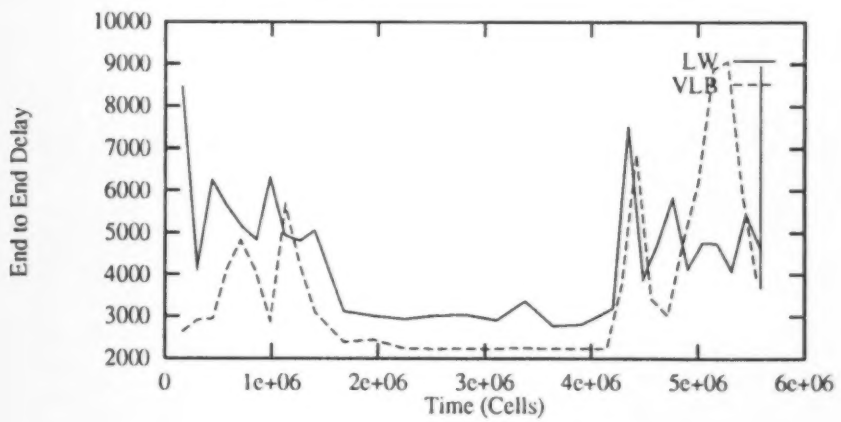

Figure 4.35: End-to-end delay variation with time for LW and VLB for B_T= 200. Load =0.8

4.3.2 Fairness.

To determine the fairness of the LW, we measure the throughput and the mean end-to-end delay in a network where a number of virtual circuits have a propagation delay of τ and others have a propagation delay of 2τ. The results presented in Tables 4.7 and 4.8 are for τ equal to 1100, and frames transmitted using strategy A. Node 1 in the tables has a propagation delay of τ, while Node 2 has a propagation delay of 2τ. The results in Tables 4.7 is for 0.4 of the offered load addressed to Node 1 and 0.3 addressed to Node 2, making the total offered load level equal to 0.7. The throughput for each of the nodes at this load is close to the offered load for both the LW and the VLB. The mean end-to-end delay for the VLB is slightly greater than the propagation delay for each of the nodes, which shows that traffic destined to each node is allocated the required capacity. The mean end-to-end delay using the LW is about twice that experienced using the VLB. This indicates that the LW has a deficiency of credits.

Table 4.8 are results for a network in which 0.4 of the offered load is destined to each of the nodes. For the VLB at this load, the throughput of Node 1 is equals 0.4 which is equal to the offered load, but the utilization level of 0.542 shows a high percentage of retransmission. Node 2 has a throughput of 0.3365 which is lower than the offered load. The average end-to-end delay shows a great disparity of transmission time for frames destined to each of the

	LW-A		VLB-A	
Source throughput	0.6906		0.715	
Buffer size	500		500	
	Node 1	Node 2	Node 1	Node 2
Utilization	0.3944	0.2962	0.4104	0.3052
Throughput	0.3897	0.2918	0.4026	0.2998
Node delay	1204	2209	1181	2279
End-to-end delay	2066	4495	1387	2639
% Retransmission	2.17%	2.3%	2.4%	2.1%

Table 4.7: Througput, delay for nodes with propagation delay τ and 2τ, $B_T = 200$ and $\rho = 0.7$

nodes. The percentage of retransmissions for Node 2 is an order of magnitude greater than that of Node 1, an indication of gross unfairness at heavy load.

Let us now take a look at the performance of the LW in Table 4.8. The throughput of Node 1 is equal to the offered load while that of Node 2 is lower than the offered load. The mean end-to-end delay however, follows a pattern similar to that in the case of a load level of 0.7, which is about twice the propagation delay for each of the nodes. The percentage of retransmitted frames to each of the nodes is of the same order of magnitude. Comparing the mean end-to-end delay and the percentage of frames retransmitted for the two node using the LW and the VLB, it is clear that the LW has a better performance in these two respects.

The results in presented in Tables 4.9 and 4.10 are obtained using strategy B. They show a similar trend as the result obtained by strategy A.

	LW-A		VLB-A	
Source throughput	0.7919		0.9645	
Buffer size	600		600	
	Node 1	Node 2	Node 1	Node 2
Utilization	0.3985	0.3935	0.5420	0.4222
Throughput	0.3929	0.3869	0.4009	0.3365
Node delay	1320	2216	1461	2432
End-to-end delay	2449	4832	14902	66469
%Retransmission	2.5%	3.2%	14.9%	77%

Table 4.8: Througput, delay for nodes with propagation delay τ and 2τ, B_T = 300 and $\rho = 0.8$

	LW-B		VLB-B	
Source throughput	0.6895		0.7267	
Buffer size	500		500	
	Node 1	Node 2	Node 1	Node 2
Utilization	0.3935	0.2960	0.4151	0.3116
Throughput	0.3896	0.2924	0.4007	0.3025
Node delay	1204	2209	1170	2265
End-to-end delay	2035	4324	1593	3041
% Retransmission	2.0%	2.1%	5.0%	4.6%

Table 4.9: Througput, delay for nodes with propagation delay τ and 2τ, distingiushed credits at B_T = 200 and $\rho = 0.7$

Source throughput	LW-B		VLB-B	
Buffer size	0.7950		0.9998	
	600		600	
	Node 1	Node 2	Node 1	Node 2
Utilization	0.3891	0.4059	0.5434	0.4563
Throughput	0.3844	0.4004	0.3928	0.3876
Node delay	1327	2217	1501	2467
End-to-end delay	2510	4926	12782	68679
%Retransmission	2.5%	3.4%	17.2%	70.4%

Table 4.10: Througput, delay for nodes with propagation delay τ and 2τ, using distinguished credits, $B_I = 300$ and $\rho = 0.8$

4.4 Summary

In this chapter, the simulation model used to test the LW mechanism was presented. The model uses a two phase generator with active and silent states whose durations are geometrically distributed. The parameters used to generate traffic by each source is the same as that observed in an Ethernet with diskless work stations. Frame transmission is implemented in terms of ATM cells. The number of frames permitted to be outstanding is controlled by the window size and the excess capacity W_e. The receiver accumulates cells of a frame until an entire frame is received. If the received frame is complete and error free, a positive acknowledgment is sent. Otherwise, a negative acknowledgment is transmitted. The destination keeps track of the sequence number of frames that have been received. A retransmission request is made for missing frames. The source accumulates traffic and adjusts the window size after receiving $2W$ acknowledgments. An additive increment with a multiplicative decrement policy is used in adjusting the window size. An estimate of the excess capacity is made every 2τ.

The first simulation results compare the performance of the LW to that of the sliding window mechanism. The results show that the LW is superior to the window mechanism, in with respects to both end-to-end delay and the probability of loss.

The behavior of the LW is investigated further and its performance com-

pared to that of the VLB, which is one of the candidate congestion control mechanisms for practical implementation. The VLB is a rate based control mechanism. The rate of transmission can be defined in terms of the number of cells or frames that may be transmitted in a given interval of time. By simulation, it was determined that the rate-based on the number of frames that may be transmitted in a given interval, has a superior performance compared to the rate based on the number of cells in a given interval.

The two methods of frame transmission proposed in Chapter 3 are simulated. The first one permits transmission of a mixture of marked and unmarked cells in a frame. The second method does not allow a mixture of cells within a single frame. Simulation results are presented for nominal load levels of 0.7 and 0.8 for both methods of frame transmission. The performance measures of interest are the probability of loss at the router, the mean end-to-end delay, and the percentage of frames retransmitted. A histogram of the duration of congestion at the router is also used to study the distribution of congestion. Similarly, a histogram of the end-to-end delay is used as a measure of delay distribution before a frame is successfully received at the destination. Simulation at a load level of 0.8 is performed for both the FCFS and the RR service discipline. At a load level of 0.7, only the RR service discipline is simulated since, at this load, congestion is not very significant. No noticeable difference would result if either of the service disciplines is used.

Other performance tests conducted are the response to load changes, and

the fairness of the LW mechanism in a network that has virtual circuits with different propagation delays.

Chapter 5

Conclusions and future research

In this thesis, a congestion control technique for HS-WANs named the Leaky Window (LW), has been proposed and its performance studied using a simulation model. The LW is a modification of the sliding window mechanism. In addition to a window of frames allowed to be outstanding by the sliding window mechanism, the LW allows a finite excess number of frames to be transmitted at any time. The excess is based on an estimate of the network load. Excess cells are marked before being transmitted into the network. Marked cells may be discarded at any congested node. The LW can therefore be categorized as a hybrid of both the preventive and reactive approaches.

Comparison between the performance of the window and the LW mech-

anisms shows that the modifications introduced into the window mechanism achieve the following objectives:

1. Faster response to network load changes through the used increased rate of transmission, therefore, bandwidth utilization is higher because of increased statistical multiplexing.

2. The marking of excess cells as eligible for discarding at congested nodes enables the router to take decisions locally when congestion occurs.

Further tests of the LW and comparing to the VLB mechanism indicate that at a load level of 0.7, the VLB has a better performance both in terms of the probability of loss and the mean end-to-end delay. The distribution of the end-to-end delay in the VLB indicates that all frames are transmitted upon generation. The LW does not exhibit this behavior which indicates that frames may be delayed at the source because of lack of sufficient credits.

At higher load levels where congestion prevails, the performance of the VLB degrades dramatically. The probability of loss and the mean end-to-end delay in the VLB are worse than that of the LW mechanism. The distribution of buffer occupancy during congestion shows that a greater number of buffers is required by the VLB compared to that required by the LW mechanism. Although the value of the maximum end-to-end delay for the two mechanisms is within the same range, the distribution of end-to-end delay shows that in the VLB, a high percentage of frames are delayed for longer periods of time

than in the LW.

From these observations, we conclude that at lower loads, there is no harm in making bandwidth available on demand as long as cells in excess of the negotiated rate are marked as eligible for discarding at congested nodes. At higher loads however, the effects of using this principle can be detrimental. The probability of discarding cells is higher, so a higher percentage of frames may have to be retransmitted. A higher percentage of retransmission results in a greater aggregate offered load, which aggravates the problem of congestion and may lead to congestion collapse. By controlling the amount of excess bandwidth, the LW achieves a better performance than the VLB at high loads.

The findings of the study of the use of different frame transmission strategies reveal a few interesting observation. Strategy A, which allows a mixture of marked and unmarked cells within a frame, does not look promising at first sight. However, comparing the trend of the probability of loss at a load level of 0.8 indicates it may have a lower bound on the number of buffers required to achieve a probability of loss of below 10^{-5} compared to strategy B, which distinguishes between excess and normal credits.

Tests performed to determine the fairness of bandwidth allocation to virtual circuits with varying propagation delays indicate that the LW allocates bandwidth more fairly than the VLB. The response to load changes of the LW is comparable to that of the VLB.

In addition to the counters required in the implementation of the sliding window mechanism. the estimation process in the LW requires a timer, a counter to keeps track of the number of acknowledgments received within the estimation interval and one more counter that keep track of the number of excess credits. This is a modest addition to the requirements of the sliding window mechanism that is justified by the performance improvement.

The LW has been shown to be a viable alternative technique for congestion control in HS-WANs. However, the algorithm presented is still far from optimal. For example, the results obtained, particularly at lower loads, indicate that there is a time lag from the instant of message arrival to the time a sufficient number of credits becomes available for transmission of the entire message in the LW. A better estimation algorithm may overcome this problem.

Further tests to determine the buffer size required to achieve a lower probability of loss than those presented in this thesis need to be carried out. Experimentation with real time traffic and other forms of traffic that do not require retransmission need to be performed.

The stability or smoothness of the rate of window size variation also needs further study. In the LW mechanism, if the window size is smaller the require capacity to for the offered load, the addition transmission capacity in form of excess credits is available. The total transmission capacity utilized is therefore modified by the offered load at the transmission time. If we observe

the total transmission capacity of a source at any time, that is the sum of window size and the excess capacity, it will vary according to the offered load. The window size variation is therefore not smooth over the entire duration of the call, but varies according to the offered load and the network conditions. Thus, the smoothness property of the window variation should be redefined to take into consideration the fact that in a high speed network carrying bursty traffic, it is more important to have the required window size at the time of occurrence of the burst, rather than a smooth variation of the window size for the entire duration of the call.

Finally, there is a modification proposed in [62] to the method of handling marked cells at the router that may have significant effect on the congestion. In that modification, if a cell of a frame is discarded, all arriving cells up to the end of that frame are discarded. A further modification to this proposal that could be implemented is to discard all the other cells of that frame already in the buffer, if there are any.

Bibliography

[1] B. W. Abeysundara and A. E. Kamal. High-speed local area networks and their performance: A survey. *ACM Computing Surveys*, 23(2):221–264, June 1991.

[2] O. Aboul-Magd, H. Gilbert, and M. Wernik. Flow and congestion control for broadband packet networks. In *Proceedings of the 13th International Teletraffic Congress (ITC)*, pages 853–858. North-Holland, June 1991.

[3] J. J. Bae and T. Suda. Survey of traffic control schemes and protocols in ATM networks. *Proceedings of the IEEE*, 79(2):170–189, February 1991.

[4] K. Bala, I. Cidon, and K. Sohraby. Congestion control for high speed packet switched networks. In *Proceedings IEEE INFOCOM Conf.*, pages 520–526, San Francisco, CA, June 1990.

[5] D. Bertsekas and R. Gallager. *Data Networks.* Prentice-Hall, Englewood Cliffs, NJ, 1987.

[6] A. Bhargava. *Intergrated Broadband Networks.* Artech House, Boston London, 1991.

[7] F. Borgonovo and L. Fratta. Policing in ATM networks: an alternative approach. In *International Teletraffic Congress, Seventh Specialist Seminar*, page 10.2. Morristown, NJ, October 1990.

[8] D. R. Cheriton. VMTP A transport protocol for the next generation of computer communication. In *Proceedings of the ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, Stowe, Vt., August 1986.

[9] D. R. Cheriton and C. L. Williamson. Loss-load curves: support for rate-based congestion control in high-speed datagram networks. In *Proceedings of the ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, 1991.

[10] D. Chiu and R. Jain. Analysis of the increase and decrease algorithm for congestion avoidance in computer networks. *Computer Networks and ISDN Systems*, 17(1):1–14, 1989.

[11] T. Y. Choi. Statistical multiplexing of burst source in an ATM network. In *Multimedia*, 1989.

[12] R. Dighe, C. J. May, and G. Ramamurthy. Congestion avoidance strategies in broadband packet networks. In *Proceedings IEEE INFOCOM Conf.*, pages 295–303 (4A.1), Bal Harbour, FL, April 1991.

[13] L. Dittmann and S. B. Jacobsen. Statistical multiplexing of identical bursty sources in an ATM network. In *Proceedings of GLOBECOM*, pages 1293–1297, Hollywood, FL, December 1988.

[14] W. A. Doeringer, D. Dykeman, M. Kaiserswerth, B. W. Meister, H. Rudin, and R. Williamson. A survey of light-weight transport protocols for high-speed networks. *IEEE Transactions on Communications*, 38(11):2025–2039, November 1990.

[15] B. T. Doshi and H. Q. Nguyen. Congestion control in ISDN frame-relay networks. Technical report, AT&T Bell Laboratories, Murray Hill, NJ, November 1988.

[16] Z. Dziong, J. Choquette, K. Liao, and L. Mason. Admission control and routing in ATM networks. *Computer Networks and ISDN Systems*, 20(1-5):189–196, December 1990.

[17] A. E. Eckberg. Generalized peakedness of teletraffic processes. In *Proceedings of the Tenth International Teletraffic Congress (ITC-10)*, Montreal, June 1983. IAC, North Holland.

[18] A. E. Eckberg, B. T. Doshi, and R. Zoccolillo. Controlling congestion in B-ISDN/ATM: Issues and strategies. *IEEE Communications Magazine*, 29(9):64–70, September 1991.

[19] K. W. Fendick, D. Mitra, I. Mitrani, M. A. Rodrigues, J. B. Seery, and A. Weiss. An approach to high-performance, high-speed data networks. *IEEE Communications Magazine*, pages 74–82, October 1991.

[20] H. J. Fowler and W. E. Leland. Local area networks traffic characteristics, with implications for broadband network congestion management. *IEEE Journal on Selected Areas in Communications*, SAC-9:1139–1149, October 1987.

[21] E. Gafni and D. Bertsekas. Dynamic control of session input rates in communication networks. In *IEEE International Military Communications Conference (MILCOM)*. Boston, MA, 1982.

[22] G. Gallassi, G. Rigolio, and L. Fratta. ATM: Bandwidth assignment and bandwidth enforcement policies. In *Proceedings of GLOBECOM*, pages 1788–1793, Dallas, Texas, November 1989.

[23] M. Gerla and L. Kleinrock. Flow control: A comparative survey. *IEEE Transactions on Communications*, 28(4):553–574, April 1980.

[24] S. J. Golestani. Congestion-free communication in broadband packet networks. In *IEEE Conference Record of the International Confer-*

ence on Communications (ICC), pages 489–494, Atlanta, Georgia, April 1990.

[25] S. J. Golestani. A stop and go queueing framework for congestion control. In *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 8–18, Philadelphia, Pennsylvania, September 1990.

[26] S. J. Golestani. Duration-limited statistical multiplexing of delay sensitive traffic in packet networks. In *Proceedings IEEE INFOCOM Conf.*, pages 323–332 (4A.4). Bal Harbour, FL, April 1991.

[27] R. Gusella. A measurement study of diskless worksation traffic on an Ethernet. *IEEE Transactions on Communications*, 38(9):1557–1568, September 1990.

[28] A. Hać. Congestion control and switch buffer allocation in high-speed networks. In *Proceedings IEEE INFOCOM Conf.*, pages 314–322 (4A.3), Bal Harbour, FL. April 1991.

[29] E. L. Hahne and R. G. Gallager. Round robin scheduling for fair flow control in data communication networks. In *Conference Record of the IEEE International Conference on Communications (ICC)*, pages 103–107, 1986.

[30] E. L. Hahne, C. R. Kalmanek, and S. P. Morgan. Fairness and congestion control on a large ATM data network with dynamically adjustable

windows. In Arne Jensen and V. B. Iversen, editors, *Teletraffic and Datatraffic in a Period of Change*, pages 867–872, Copenhagen, Denmark, June 1991. North-Holland.

[31] M. Hirano and Naoya Watanabe. Characteristics of a cell multiplexer for bursty ATM traffic. In *IEEE Conference Record of the International Conference on Communications (ICC)*, pages 399–403 (13.2), Boston, MA, June 1989.

[32] J. Y. Hui. A broadband packet switch for intergrated transport. *IEEE Journal on Selected Areas in Communications*, SAC-5:1264–1273, October 1987.

[33] V. Jacobson. Congestion avoidance and control. *ACM Computer Communication Review*, 18(4):314–329, August 1988. Proceedings of the Sigcomm '88 Symposium in Stanford, CA, August, 1988.

[34] J. M. Jaffe. Bottleneck flow control. *IEEE Transactions on Communications*, 29(7):954–962, July 1981.

[35] R. Jain. A timeout-based congestion control scheme for window flow-controlled networks. *IEEE Journal on Selected Areas in Communications*, SAC-4(7):1162–1167, October 1986.

[36] R. Jain and K. K. Ramakrishnan. Congestion avoidance in computer networks with a connectionless network layer: Concepts. In *Proceedings*

of the Computer Networking Symposium, pages 134–143, Washington, DC, April 1988.

[37] T. Kamitake and T. Suda. Evaluation of an admission control scheme for an ATM network considering fluctuations in cell loss rate. In *Proceedings of GLOBECOM*, pages 1774–1780, Dallas, Texas, November 1989.

[38] N. F. Maxemchuk and M. El Zarki. Routing and flow control in high-speed wide-area networks. *Proceedings of the IEEE*, 78(1):204–221, January 1990.

[39] S. E. Minzer. Broadband ISDN and Asynchronous Transfer Mode (ATM). *IEEE Communications Magazine*, pages 17–24, September 1989.

[40] D. Mitra. Optimal design of congestion control for high speed data networks. Technical report, AT&T Bell Laboratories, Murray Hill, NJ, September 1989.

[41] J. F. Mollenauer. Standards for metropolitan area networks. *IEEE Communications Magazine*, 26(4):15–19, April 1988.

[42] José Augusto Suruagy Monteiro, Mario Gerla, and Luigi Fratta. Input rate control for ATM networks. In J. W. Cohen and Charles D. Pack, editors, *Queueing, Performance and Control in ATM — Proceedings of the Workshop at the 13th International Teletraffic Congress (ITC)*,

pages 117–122, Copenhagen, Denmark, June 1991. North-Holland. Volume 15 of the North Holland Studies in Telecommunication.

[43] K. Nakamaki, M. Kawakatsu, and A. Notoya. Traffic control for ATM networks. In *IEEE Conference Record of the International Conference on Communications (ICC)*, pages 713–717 (53.1), Boston, June 1989.

[44] A. N. Netravali, W. D. Roome, and K. Sabnani. Design and implementation of a high-speed transport protocol. *IEEE Transactions on Communications*, 38(11):2010–2024, November 1990.

[45] C. Partridge. *Innovations in Internetworking*. Artech House, Boston London, 1988.

[46] D. W. Petr and V. S. Frost. Optimal packet discarding: An ATM-oriented analysis model and initial results. In *Proceedings IEEE INFOCOM Conf.*, pages 537–542, San Francisco, CA, June 1990.

[47] K. K. Ramakrishnan, D. M. Chiu, and R. Jain. Congestion avoidance in computer networks with a connectionless network layer, Part IV selective binary feeedback scheme for general topologies. Technical Report TR-509, Digital quipment Corporation, 1987.

[48] K. K. Ramakrishnan and R. Jain. A binary feedback scheme for congestion avoidance in computer networks with a connectionless network

layer. In *Proceedings of the 1988 SIGCOMM Symposium on Communications Architectures and Protocols*, pages 303–313, Stanford, CA, August 1988. ACM.

[49] G. Ramamurthy and R. S. Dighe. Distributed source control: A network access control for intergrated broadband packet networks. In *Proceedings IEEE INFOCOM conf.*, pages 896–907, April 1990.

[50] E. P. Rathgeb and T.H. Theimer. The policing function in ATM networks. In *Proceedings of International Switching Symposium*, pages 127–130, May 1990.

[51] G. Rigolio and L. Fratta. Input rate regulation and bandwidth assignment in ATM networks: An integrated approach. In J. W. Cohen and Charles D. Pack, editors, *Queueing, Performance and Control in ATM — Proceedings of the Workshop at the 13th International Teletraffic Congress (ITC)*, pages 123–128, Copenhagen, Denmark, June 1991. North-Holland. Volume 15 of the North Holland Studies in Telecommunication.

[52] H. Robbin and S. Monro. A stochastic approximation method. In *Ann. Math. Stat.*, volume 22, pages 400–407, 1951.

[53] F. E. Ross. FDDI–A tutorial. *IEEE Communications Magazine*, 24(5):10–17, May 1986.

[54] H. Schulzrinne, J. F. Kurose, and D. Towsley. Congestion control for real-time traffic in high-speed networks. In *Proceedings IEEE INFOCOM Conf.*, pages 543–550, San Francisco, CA, June 1990.

[55] M. Schwartz. *Telecommunication Networks: Protocols, Modeling and Analysis*. Addison-Wesley, Reading, MA, 1987.

[56] M. Sidi, W. Liu, I. Cidon, and I. Gopal. Congestion control through input rate regulation. In *Proceedings of GLOBECOM*, pages 1764–1768, Dallas, Texas, November 1989.

[57] K. Sriram and D. M. Lucantoni. Traffic smoothing effects of bit dropping in a packet voice multiplexer. In *Proceedings IEEE INFOCOM Conf.*, pages 759–769, New Orleans, March 1988.

[58] K. Sriram and W. Whitt. Characterizing superposition arrival processes in packet multiplexers for voice and data. In *Proceedings of GLOBECOM*, pages 778–784 (25.4), New Orleans, LA, December 1985.

[59] W. Stallings. *Handbook of computer commuications:Local network standards*, volume 2. Howard W. Sams and Company, Indianapolis, Indiana, 1st edition, 1987.

[60] CCITT (International Telegraph Study Group XVIII and Telephone Consultative Committee). Draft recommendation I.1361: ATM

layer specification for B-ISDN, January 1990. Appendix 2 to Annex 1 of Question 24/XV (COM XV-1-E).

[61] CCITT (International Telegraph Study Group XVIII and Telephone Consultative Committee). Draft recommendation I.150: Sec. 3.4.2.3.2, January 1990. Appendix 2 to Annex 1 of Question 24/XV (COM XV-1-E).

[62] Lyon T. Simple and efficient adaptation layer (SEAL), 1991. Standards Project T1S1.5AAL.

[63] A. S. Tanenbaum. *Computer Networks*. Prentice Hall, Englewood Cliffs, NJ, 2nd edition, 1988.

[64] Z. Wang and J. Crowcroft. A new congestion control scheme: Slow start and search (Tri-S). *ACM Computer Communication Review*, 21(1):32–43, January 1991.

[65] N. Yin, S. Li, and T. E. Stern. Congestion control for packet voice by selective packet discarding. In *Proceedings of GLOBECOM*, pages 1782–1786, Tokyo, November 1987.

[66] C. Yuan and J. A. Silvester. Queueing analysis of delay constrained voice traffic in a packet switching system. *IEEE Journal on Selected Areas in Communications*, SAC-7(5):729–738, June 1989.

[67] L. Zhang. VirtualClock: A new traffic control algorithm for packet switching networks. In *ACM SIGCOMM Symposium on Communications Architectures and Protocols*, pages 19–29, Philadelphia, PA, September 1990.